UNCLASSIFIED

AD NUMBER

AD881214

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; Nov 1970. Other requests shall be referred to Office of Naval Research, Washington, D.C.

AUTHORITY

ONR ltr, 29 Aug 1973

THIS PAGE IS UNCLASSIFIED

AD No. ____
DDC FILE COPY

AD881214

AD881214

SIMULATION STUDY OF A MODULAR ESCAPE CAPSULE

RELEASED FROM A SUBMERGED SUBMARINE

J. C. Gebhardt
F. A. Klappenberger

CADCOM, Incorporated
2024 West Street
Annapolis, Md. 21401

This research was sponsored by the Office of Naval Research under Contract N00014-70-C-0247/NR277-061/11-25-69.

November 1970
Final Report

Prepared for

Office of Naval Research
Washington, D. C.
Under Contract Number N00014-70-C-0247

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Office of Naval Research.

Preface

This report describes the development of a real-time digital simulation of the motions of a modular escape capsule released from a submerged, distressed submarine. The work was performed for the Office of Naval Research under Contract Number N00014-70-C-0247.

The authors are indebted to many people for their substantial contributions to this project. Drs. Adams, Carson, and Rogers performed a great deal of the analysis, while Mr. P. Gann did a great deal of the computer programming. Mrs. Sandy Green deserves a special thank you for rapid accurate typing of the manuscript. The advice and guidance of Mr. B. Friedman of ONR is greatly appreciated.

Table of Contents

Table of Contents (Continued)

LIST OF FIGURES

I. INTRODUCTION

The modern, nuclear submarine is one of the most effective weapon systems in use today but it suffers from one major flaw -- in case of disaster the crew has very little chance of survival. Future submarines will be able to operate deeper and deeper and hence the problem cannot help but become more critical.

In the aircraft industry, the methods of crew escape have evolved from the parachute, to the ejection seat, to the modular ejection capsule. The parallels to this pattern for crew escape from submarines are illustrated below:

Development Stage	Aircraft	Submarine
A. Early Development:	no escape provision	no escape provision
B. Early Operational Vehicles:	Parachute	Momson Lung (Free Ascent)
C. Recent Operational Vehicles:	Ejection Seat	Steinke Hood
D. Recent Developments:	Escape Capsule	?

The diving bell and the DSRV have no parallels in aircraft escape and are not applicable unless the men <u>survive</u> the emergency. This requires operation in water depths less than the crush depth of the vehicle. In addition, free ascent is limited to relatively shallow depths as is illustrated in Figure I-1.

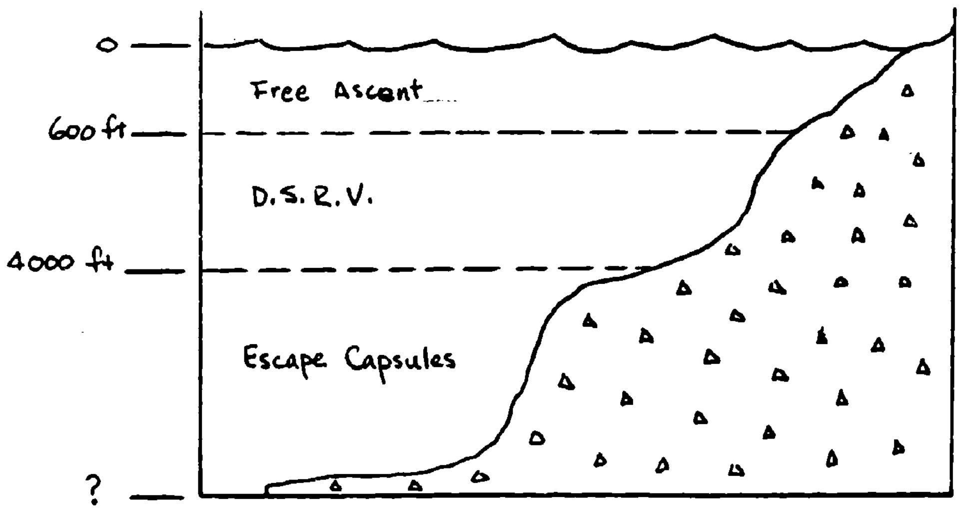


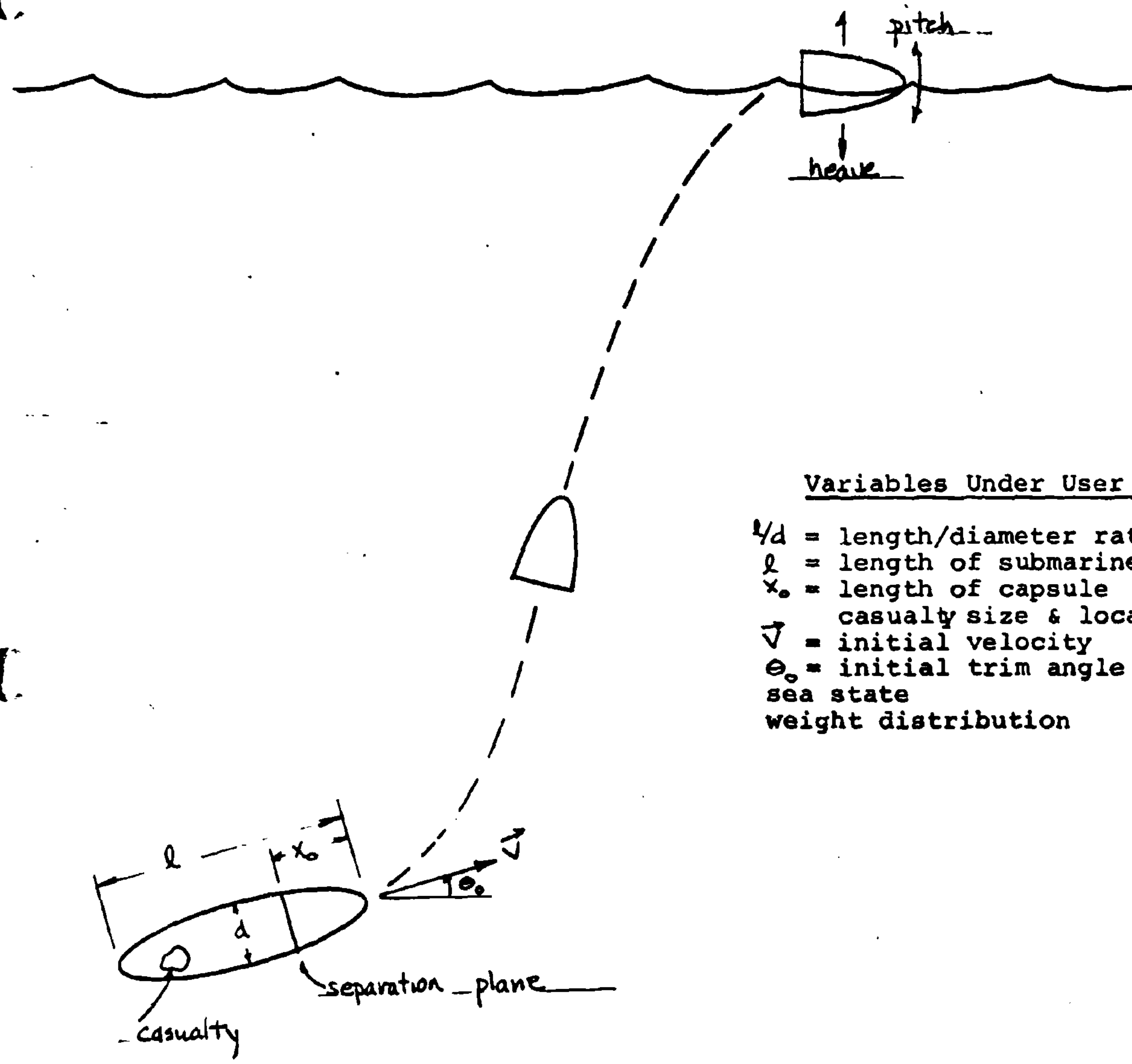
Figure I-1. Depth Limits for Various Rescue Techniques

Thus, it is quite logical that the next step in the development of escape systems for distressed submarines involves a direct means of escape utilizing a modular escape capsule.

The interface requirements with the submarine involve many questions: Can the submersible be controlled from the capsule? What is the arrangement of men and materials inside the capsule? How is the module separated from the submersible? What are the static and dynamic stabilities of the complete submersible and of the separated capsule? The answers to these questions will greatly influence the size and shape of the escape module; but, there are also considerations in the ascent to the surface and in the survival requirements on the surface which will influence the size and shape of the capsule. An operational capsule will therefore represent a compromise between the various requirements for submarine interface, ascent requirements, and surface survival requirements.

In the present study, the hydrodynamic behavior of an escape capsule during the separation and ascent phase has been analysed and simulated in a real-time digital system called MODSEC - Modular Submarine with Escape Capsule. The MODSEC system can simulate the behavior of a submarine with a forward escape capsule as the submarine sustains damage, the capsule separates, ascends to the surface, and floats on the surface awaiting rescue. Figure I-2 illustrates, in a condensed fashion, the capabilities of the simulation.

The MODSEC system has been designed as a real-time system utilizing a time-shared computer. A Typagraph Model 3 terminal is used as input/output device. This terminal allows the user to obtain high quality graphical output very quickly. The MODSEC system incorporates a comprehensive command language which enables the user to change input parameters, select output, and analyze various phases of the capsule motions at his option. Disc file storage is incorporated into the system to allow the user to store lengthy weight distributions permanently. This feature prevents costly data loss due to computer crashes, communications failure, or user mistakes. The MODSEC system has been designed so that personnel untrained in the computer sciences can easily use the system. No knowledge of computer languages is required and no tedious, complicated, data entry is necessary.



Variables Under User Control

ℓ/d = length/diameter ratio
ℓ = length of submarine
x_o = length of capsule
casualty size & location
$\vec{V}$ = initial velocity
θ_o = initial trim angle
sea state
weight distribution

Figure I-2. MODSEC Simulation.

The MODSEC system is presently a sophisticated interactive system for investigating the hydrodynamic behavior of truncated ellipsoids of revolution. Because of its modular design, MODSEC has great potential as a tool for the design of underwater or flying bodies of virtually any geometry moving in up to 6 degrees of freedom. It is an easy job to change the equations of motion routine or the geometry routine to accomodate other geometries and environments.

II. PRELIMINARY CONSIDERATIONS

The analysis of a vehicle having six major degrees of freedom as well as numerous internal degrees of freedom caused by elastic, gyroscopic, and control effects is usually approached in the following manner:

(a) The vehicle is first regarded as a rigid body and the equations of motion are derived with respect to a set of axes fixed in it. These are Euler's equations.

(b) Modifications required by gyroscopic, elastic and control effects are evaluated.

(c) Due to moments and products of inertia appearing in Euler's equations, a choice must be made as to whether the coordinate system be earth fixed, in which case these terms become functions of body orientation; or whether body axes are to be adopted, which avoids variable moments and products of inertia, but introduces the difficulty that a complicated coordinate transformation must be introduced to relate body and space fixed axes. The latter alternative is usually adopted.

(d) The resulting trajectory equations are simultaneous integro-differential equations, and cannot be conveniently solved without extensive linearization and approximations.

For the class of problems under consideration here, however, the trajectory analysis is considerably simplified since the motion takes place in a plane, and the longitudinal axes of the vehicle remains in the plane. Thus there is no coordinate variation of moment of inertia since the only moment of inertia relevant to the problem is that which lies about the axis normal to the plane of motion. No elastic, gyroscopic or control effects are to be considered, and then the motion may be referred to a single set of earth fixed axes by a simple transformation.

This is not to say, however, that the resulting equations are easily solved: for the essence of this problem is the inclusion of large angular displacements, which prohibits linearizations to a large extent. Further, the effects of induced mass and moments of inertia must be included and finally, the modeling of the various hydrodynamic forces and moments for large angular displacements requires considerable engineering judgement.

III. EQUATIONS OF MOTION

Consider a rigid body (either the escape capsule or the remaining submarine) moving in an earth fixed x, y coordinate system. It can be shown that bodies with axial symmetry moving under the influence of hydrodynamic, hydrostatic, gravitational, and inertial forces will move in a single plane. The problem is now reduced from a general 6 degree of freedom problem to one with 3 degrees of freedom. The body motion can be completely described by the x, y coordinates of the center of mass and the angle θ , expressed as a function of time. See Figure III-1.

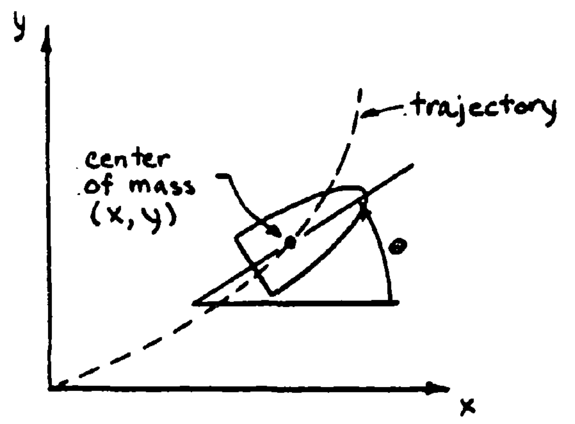


Figure III-1. Coordinate System

The forces on the body must be expressed as functions of the geometry of the body, its velocity, and its orientation. These forces are of three types: hydrodynamic, hydrostatic, and inertial.

Referring to Figure III-2 we assume that the lift force, L, and the drag force, D, act through the hydrodynamic center h, the lift force perpendicular to the velocity vector $\vec{V}$, the drag force parallel to $\vec{V}$.

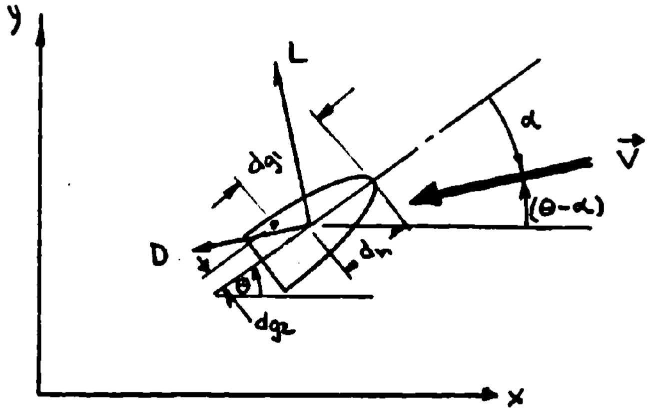


Figure III-2.

Resolving the lift and drag into their x and y components we have

$$F_{hx} = -L \sin(\theta-\alpha) - D \cos(\theta-\alpha)$$

$$F_{hy} = L \cos(\theta-\alpha) - D \sin(\theta-\alpha)$$

where F_{hx} and F_{hy} are the x and y components of the hydrodynamic force. The moment of the hydrodynamic force about the center of the mass, g, can be written (counter clockwise moments positive)

$$M_h = -F_{hx}\left[(d_{g1}-d_h)\sin\theta - d_{g2}\cos\theta\right] + F_{hy}\left[(d_{g1}-d_h)\cos\theta + d_{g2}\sin\theta\right] - M_d$$

where M_d is the hydrodynamic damping moment.

It can be shown that $M_d = K_3\,\dot{\theta}^2$ where

$$K_3 = \frac{1}{2}\rho C_s \int_0^{x_1} y(x)\,(d_{g1}-x)^2\,dx$$

In the above expression, C_s is the crossflow drag coefficient for a circular cylinder and $y(x)$ is the equation of the ellipsoid in the coordinate system of Section IV.

The hydrostatic forces include the buoyant force acting upward on the vehicle through the centroid of the underwater volume, b, and the weight of the vehicle acting downward through the center of mass (See Figure III-3).

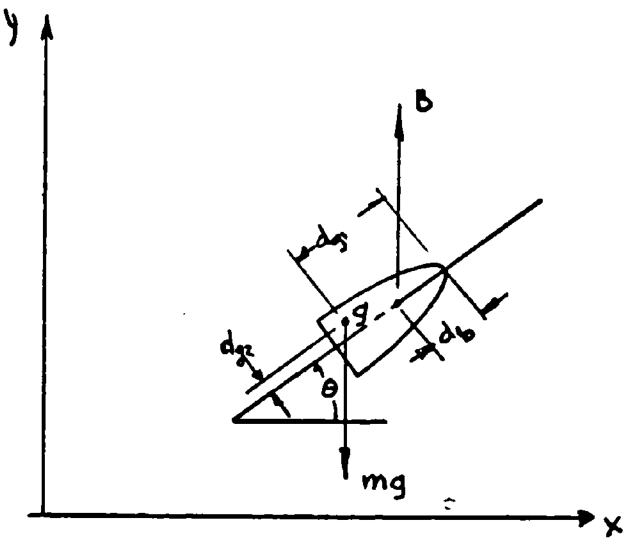


Figure III-3. Hydrostatic and Buoyant Forces.

Resolving the hydrostatic forces into their x and y components we get

$$F_{sx} = 0$$

$$F_{sy} = B - mg$$

Taking moments about g we get

$$M_s = B\left[(d_{g1} - d_b)\cos\theta + d_{g2}\sin\theta \right]$$

The inertial forces and moment are proportional to the accelerations and can be written

$$F_{ix} = -(m + m_{vx})\ddot{x}$$

$$F_{iy} = -(m + m_{vy})\ddot{y}$$

$$M_i = -(I + I_v)\ddot{\theta}$$

where m_{vx}, m_{vy}, I_v are the virtual masses and moment of inertia.

The 3 equations of motion can now be written

$$F_{hx} + F_{sx} + F_{ix} = 0$$
$$F_{hy} + F_{sy} + F_{iy} = 0$$
$$M_h + M_s + M_i = 0$$

or

$$(m + m_{vx})\ddot{x} = -L\sin(\theta-\alpha) - D\cos(\theta-\alpha)$$
$$(m + m_{vy})\ddot{y} = B - mg + L\cos(\theta-\alpha) - D\sin(\theta-\alpha)$$
$$(I + I_v)\ddot{\theta} = B[(d_{g1}-d_b)\cos\theta + d_{g2}\sin\theta] + (D\sin\alpha + L\sin\alpha)(d_{g1}-d_h) + (L\sin\alpha - D\cos\alpha)\,d_{g2} - M_d$$

Non-dimensionalizing equations 1, 2, and 3 by non-dimensionalizing all lengths by the body length, l, and writing

$$\vec{V}^* = \vec{v}/\sqrt{g\ell}, \quad t^* = \sqrt{g/\ell}\; t$$
$$B^* = \frac{2B}{\rho g \ell S}, \quad m^* = \frac{2m}{\rho \ell S}$$
$$I^* = \frac{2I}{\rho \ell^3 S}, \quad \text{and}$$
$$P = \frac{B/\rho g}{\ell S} = \text{prismatic coefficient}$$

and introducing the lift and drag coefficients

$$C_L \frac{L}{\frac{1}{2}\rho V^2 S}, \quad C_D \frac{D}{\frac{1}{2}\rho V^2 S}$$

we get the non-dimensional equations of motion for x*, y*, and θ.

$$m_x'^*\frac{d^2x^*}{dt^{*2}} = -\left[\left(\frac{dx^*}{dt^*}\right)^2 + \left(\frac{dy^*}{dt^*}\right)^2\right]\cdot\left[C_L\sin(\theta-\alpha) + C_D\cos(\theta-\alpha)\right]$$

$$m_y'^*\frac{d^2y^*}{dt^{*2}} = 2P - m^* + \left[\left(\frac{dx^*}{dt^*}\right)^2 + \left(\frac{dy^*}{dt^*}\right)^2\right]\cdot\left[C_L\cos(\theta-\alpha) - C_D\sin(\theta-\alpha)\right]$$

$$I'^*\frac{d^2\theta}{dt^{*2}} = \left[\left(\frac{dx^*}{dt^*}\right)^2 + \left(\frac{dy^*}{dt^*}\right)^2\right]\cdot\left[(C_D\sin\alpha + C_L\cos\alpha)(d_{g1}^* - d_h^*)\right.$$

$$\left. + (C_L\sin\alpha - C_D\cos\alpha)\,d_{g2}^*\right] + 2P\left[(d_{g1} - d_b)\cos\theta + d_{g2}\sin\theta\right] - K_3\dot{\theta}|\dot{\theta}|$$

If we now assume that the virtual masses can be written

$$m_{vx} = \rho \nabla k_x$$

$$m_{vy} = \rho \nabla k_y$$

$$I_v = \rho \nabla r_g^2 K_\theta$$

Then

$$m_x'^* = m^* + 2P k_x$$

$$m_y'^* = m^* + 2P k_y$$

and

$$I'^* = I^* + 2P r_g^{*2} K_\theta$$

where K_x, K_y, and K_θ are the virtual mass and inertia coefficients in the x, y, and Θ directions. These coefficients are discussed in more detail in Section V-2.

The first step in the solution of these three coupled, second order, ordinary, differential equations is to obtain an equivalent set of six first order, differential equations. This is accomplished as follows:

Let

$$Y_1 = x^*$$

$$Y_2 = y^*$$

$$Y_3 = \theta$$

$$Y_4 = dx^*/dt^*$$

$$Y_5 = dy^*/dt^*$$

$$Y_6 = d\theta/dt^* \qquad \text{and } X = t^*$$

Then define:

$$F_1 = dY_1/dX = Y_4$$

$$F_2 = dY_2/dX = Y_5$$

$$F_3 = dY_3/dX = Y_6$$

$$F_4 = dY_4/dX = d^2x^*/dt^{*2}$$

$$F_5 = dY_5/dX = d^2y^*/dt^{*2}$$

$$F_6 = dY_6/dX = d^2\theta/dt^{*2}$$

The preceding six equations are six first order equations with six dependent variables (y_1, y_2, --- y_6) and one independent variable (x). Initial conditions for the six dependent variables are set by specifying values of $y_1(0)$, $y_2(0)$, --- $y_6(0)$.

We use a fourth order Runge-Kutta technique to solve simultaneously the six first order differential equations of the form

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, y_3 \cdots y_n)$$
$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, y_3, \cdots y_n)$$
$$\vdots$$
$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, y_3, \cdots y_n)$$

The algorithm used to calculate $y_i(x+h)$ given $y_i(x)$ is

$$y_i(x+h) = y_i(x) + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

$$K_1 = f_i(x, y_1, y_2, \cdots y_n)$$

$$K_2 = f_i(x+\tfrac{1}{2}h, y_1+\tfrac{1}{2}hK_1, y_2+\tfrac{1}{2}hK_1, y_3+\tfrac{1}{2}hK_1, \cdots y_n+\tfrac{1}{2}hK_1)$$

$$K_3 = f_i(x+\tfrac{1}{2}h, y_1+\tfrac{1}{2}hK_2, y_2+\tfrac{1}{2}hK_2, y_3+\tfrac{1}{2}hK_2, \cdots y_n+\tfrac{1}{2}hK_2)$$

$$K_4 = f_i(x+h, y_1+hK_3, y_2+hK_3, y_3+hK_3, \cdots y_n+hK_3)$$

where h is the non-dimensional time increment.

IV. SIMULATED VEHICLE GEOMETRY AND WEIGHT DISTRIBUTION

A thorough investigation of representative submarine hull shapes revealed that the intact vehicle can be mathematically represented by an ellipsoidal body of revolution. This selection provides a reasonably accurate geometric model which is relatively simple to manipulate mathematically. Although some inaccuracies are introduced in the after portion of the submarine, the study is primarily concerned with the forward section -- the capsule -- which is very closely approximated by a portion of an ellipsoidal body of revolution. Figure IV-1 illustrates the plan view, with the appropriate axes.

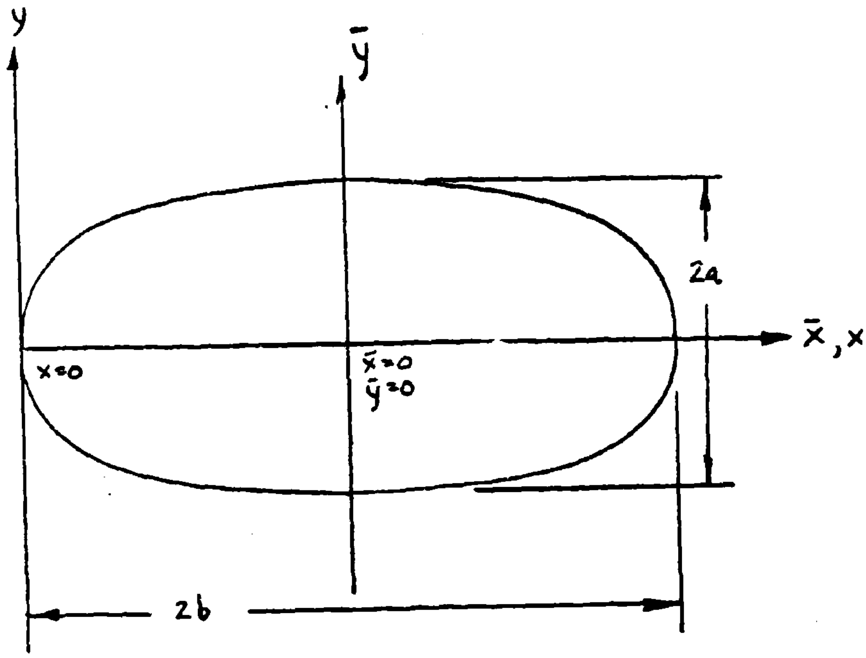


Figure IV-1. Ellipsoidal Body of Revolution

Vehicle Geometry

It is assumed that the major axis of the ellipsoid is the axis of revolution and that this corresponds to the longitudinal reference axis of the vehicle. The equation of an ellipse in the $\bar{x}$, $\bar{y}$ axis system (Fig. IV-1) is

$$\frac{\bar{x}^2}{b^2} + \frac{\bar{y}^2}{a^2} = 1$$

where a is the semi-minor axis, and b is the semi-major axis. Noting that

$$x = \bar{x} + b$$

$$y = \bar{y}$$

the equation of the ellipse in the x, y coordinate system is

$$\frac{(x-b)^2}{b^2} + \frac{y^2}{a^2} = 1 \tag{1}$$

The geometric quantities that are of interest in the analysis of the dynamics of the vehicle are:

volume,
surface area,
volume moment of inertia,
radius of gyration,
longitudinal center of gravity of the vehicle shell,
longitudinal center of buoyancy.

The geometry subroutine calculates each of these parameters for the intact vehicle, the capsule, and the remaining vehicle.

Volume

The volume of any segment of the ellipsoid of revolution, say between stations x_1 and x_2, is obtained from

$$\nabla = \int_{x_1}^{x_2} \pi y^2 \, dx \tag{2}$$

Solving Eq. (1) for y^2:

$$y^2 = \left[1 - \frac{(x-b)^2}{b^2}\right] a^2 \ . \tag{3}$$

Substituting this into Eq. (2):

$$\nabla = \int_{x_1}^{x_2} \pi \left[1 - \frac{(x-b)^2}{b^2}\right] a^2 dx$$

Integration yields:

$$\nabla(x_2, x_1) = \frac{\pi}{3}\left(\frac{L}{D}\right)^{-2}\left[3(x_2 - x_1)\, b^2 + (x_1 - b)^3 - (x_2 - b)^3\right] ,$$

where the length-to-diameter ratio was taken as L/D = b/a. Non-dimensionalizing the volume with respect to the length:

$$\nabla^* = \frac{\nabla}{L^3} = \frac{\pi}{24}\left(\frac{L}{D}\right)^{-2}\left[3(x_2^* - x_1^*) + (x_1^* - 1)^3 - (x_2^* - 1)^3\right]$$

where $x_2^* = x_2/b$

$x_1^* = x_1/b$

Surface Area

The surface area of any segment of an ellipsoid of revolution between stations x_1 and x_2 is given by:

$$A = 2\pi \int_{x_1}^{x_2} y \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$

Differentiating Equation (3):

$$\frac{dy}{dx} = \left(\frac{a}{b}\right) \frac{(b-x)}{(2bx - x^2)^{1/2}} \tag{4}$$

From Eqs. (3) and (4):

$$A = 2\pi \int_{x_1}^{x_2} \frac{a}{b}\left[a^2 + 2b\left(1 - \frac{a^2}{b^2}\right)x - \left(1 - \frac{a^2}{b^2}\right)x^2\right]^{1/2} dx \tag{5}$$

Integrating Eq. (5):

$$A = 2\pi \frac{a}{b} \left\{ \frac{(x-b)}{2} \left[a^2 + 2b\left(1-\frac{a^2}{b^2}\right)x - \left(1-\frac{a^2}{b^2}\right)x^2 \right]^{1/2} \right.$$

$$\left. + \frac{b^2}{2} \frac{1}{\left(1-\frac{a^2}{b^2}\right)^{1/2}} \sin^{-1}\left[\left(\frac{x-b}{b}\right)\left(1-\frac{a^2}{b^2}\right)^{1/2} \right] \right\}_{x_1}^{x_2} \qquad (5a)$$

Noting that as a → b (or a/b → 1), the second term involving the Arcsin becomes indeterminate, we use the fact that as $x \to 0$, $\sin^{-1} x \approx x$

Thus from Equation (5a), as $a/b \to 1$,

$$A \approx 2\pi\left(\frac{a}{b}\right) \left\{ \frac{(x-b)}{2} \left[a^2 + 2b\left(1-\frac{a^2}{b^2}\right)x - \left(1-\frac{a^2}{b^2}\right)x^2 \right]^{1/2} \right.$$

$$\left. + \frac{b^2}{2} \frac{(x-b)}{b} \right\}_{x_1}^{x_2}$$

Returning to the full equation and non-dimensionalizing,

$$A^* = \frac{A}{L^2} = \frac{\pi}{2}\left(\frac{L}{D}\right)^{-1} \left\{ \frac{(x^*-1)}{2} \left[\left(\frac{D}{L}\right)^2 + 2\left[1-\left(\frac{D}{L}\right)^2\right]x^* - \left[1-\left(\frac{D}{L}\right)^2\right]x^{*2} \right]^{1/2} \right.$$

$$\left. + \frac{1}{2\left[1-\left(\frac{D}{L}\right)^2\right]^{1/2}} \sin^{-1}\left[(x^*-1)\left(1-\left(\frac{D}{L}\right)^2\right)^{1/2} \right] \right\}_{x_1^*}^{x_2^*}$$

Volume Moment of Inertia

The volume moment of inertia (I) of any segment of the ellipsoid of revolution about the point x=0 is given by:

$$I = \int_{x_1}^{x_2} \rho x^2 \, d\nabla$$

where ρ is the density of the ellipsoid. The differential

volume of a slice of the ellipsoid is $d\nabla = \pi y^2 dx$ and, hence, the moment of inertia:

$$I = \rho \int_{x_1}^{x_2} \pi y^2 x^2 dx$$

Using Eq. (3) allows writing this as

$$\frac{I}{\rho} = \pi a^2 \int_{x_1}^{x_2} \left[1 - \frac{(x-b)^2}{b^2}\right] x^2 dx$$

Integration yields:

$$\frac{I}{\rho} = \frac{\pi a^2 b^3}{10} \left\{ 5\left[\left(\frac{x_2}{b}\right)^4 - \left(\frac{x_1}{b}\right)^4\right] - 2\left[\left(\frac{x_2}{b}\right)^5 - \left(\frac{x_1}{b}\right)^5\right] \right\} ,$$

which, upon non-dimensionalizing with respect to the vehicle length (2b) to the fifth power, yields:

$$\frac{I}{\rho L^5} = \frac{\pi}{320} \left(\frac{L}{D}\right)^{-2} \left\{ 5\left(x_2^{*4} - x_1^{*4}\right) - 2\left(x_2^{*5} - x_1^{*5}\right) \right\}$$

Using the parallel - axis theorem allows calculation of the moment of inertia about any axis if the distance from x=0 to the axis is known; i.e.,

$$I_2 = I_{cg} + (\text{mass})(\text{distance between axes})^2$$

As an example, the moment of inertia about the center of gravity is

$$\frac{I}{\rho} = \frac{I_{cg}}{\rho} + \frac{m}{\rho} x_{cg}^2$$

or

$$\frac{I_{cg}}{\rho} = \frac{I}{\rho} - \frac{m}{\rho} x_{cg}^2 \qquad (6)$$

Radius of Gyration

The radius of gyration (r_g) is given by

$$r_g = \frac{I}{m} \qquad (7)$$

Thus, from the previous results:

$$r_g = \sqrt{\frac{I/\rho}{\nabla}}$$

The value may be obtained from previously calculated results. The radius of gyration may also be transferred to any other axis using the parallel-axis system. From Eqs. (6) and (7):

$$(r_g)^2_{cg} = (r_g)^2_{x=0} - X^2_{cg}$$

Longitudinal Center of Bouyancy

The longitudinal center of buoyancy of any segment of the ellipsoid of revolution is given by the first moment of volume about the point x=0, divided by the volume:

$$X_{LCB} = \frac{\int_{x_1}^{x_2} \pi x y^2 dx}{\nabla(x_1, x_2)} \qquad (8)$$

Using Eq. (3), the integral in the numerator of Eq. (8) may be written:

$$\pi \int_{x_1}^{x_2} x y^2 dx = \pi a^2 \int_{x_1}^{x_2} x \left[1 - \frac{(x-b)^2}{b^2} \right] dx$$

Integration yields:

$$\pi \int_{x_1}^{x_2} x y^2 dx = \pi a^2 \left[\frac{2(x_2^2 - x_1^2)}{3b} - \frac{(x_2^4 - x_1^4)}{4b^2} \right]$$

Thus, the longitudinal center of buoyancy of any segment of the ellipse:

$$X_{LCB} = \frac{\frac{2(x_2^3 - x_1^3)}{3b} - \frac{(x_2^4 - x_1^4)}{4b^2}}{\frac{1}{3b^2}\left[3(x_2 - x_1) b^2 + (x_1 - b)^3 - (x_2 - b)^3 \right]}$$

Non-dimensionalizing with respect to the length (2b):

$$X^*_{LCB} = \frac{X_{LCB}}{2b} \quad \frac{\frac{2}{3}\left(x_2^{*3} - x_1^{*3}\right) - \frac{1}{4}\left(x_2^{*4} - x_1^{*4}\right)}{\frac{2}{3}\left[3\left(x_2^* - x_1^*\right) + \left(x_1^* - 1\right)^3 - \left(x_2^* - 1\right)^3\right]}$$

Centroid of the Projected Area

The centroid of the projected area of a segment of an ellipsoid of revolution can be written

$$X_p = \frac{\int_{x_1}^{x_2} x y \, dx}{\int_{x_1}^{x_2} y \, dx}$$

With the help of Equation (3) this can be written

$$X_p = \frac{\int_{x_1}^{x_2} x\left[1 - \frac{(x-b)^2}{b^2}\right]^{1/2} dx}{\int_{x_1}^{x_2} \left[1 - \frac{(x-b)^2}{b^2}\right]^{1/2} dx}$$

Carrying out the integrations in both the numerator and denominator and non-dimensionalizing as before, yields

$$x_p = \frac{-\frac{1}{3}\left[2x^* - x^{*2}\right]^{3/2} + \frac{1}{2}\left[(x^*-1)\left(2x^* - x^{*2}\right)^{1/2} + \sin^{-1}(x^*-1)\right]\Big|_{x_1^*}^{x_2^*}}{\frac{1}{2}\left[(x^*-1)\left(2x^* - x^{*2}\right)^{1/2} + \sin^{-1}(x^*-1)\right]\Big|_{x_1^*}^{x_2^*}}$$

Longitudinal Center of Gravity of the Shell

The longitudinal center of gravity of the capsule and remaining vehicle shell is required for the subsequent determination of the center of gravity of the intact vehicle, the capsule, and/or the remaining vehicle. To do this the total weight of the vehicle shell must first be calculated. We assume that the capsule and the remaining vehicle have constant, but not necessarily equal, shell thickness and may even be constructed of different materials. Since the shell thickness is small compared to the average radius of curvature, the weight of the capsule shell may be closely approximated by the surface area of the shell times the shell thickness times the specific weight of the shell material, and, similarly, for the remaining vehicle. The total shell weight for the intact vehicle is then the sum of these two weights. Note that these are the ellipsoidal shell weights and do not include the weight of any end plate required to close off the capsule and remaining vehicle from the surrounding environment upon separation.

The longitudinal center of gravity of any segment of the ellipsoidal shell can now be written as

$$c.g._{shell} = \frac{2\pi t_o w_o \int_{x_1}^{x_2} xy\sqrt{1+\left(\frac{dy}{dx}\right)^2}\,dx}{(\text{total weight})}$$

From the above discussion, the total weight of the shell is $t_0 w_0 S$. Thus, the center of gravity becomes:

$$c.g._{shell} = \frac{2\pi}{S}\int_{x_1}^{x_2} xy\sqrt{1+\left(\frac{dy}{dx}\right)^2}\,dx$$

Evaluation of the integral yields:

$$c.g._{shell} = \left.\frac{\left[a^2 + 2b\left(1-\frac{a^2}{b^2}\right)x - \left(1-\frac{a^2}{b^2}\right)x^2\right]^{3/2}}{3\left[\frac{a^2}{b^2}-1\right]\left[\frac{S}{2\pi\left(\frac{a}{b}\right)}\right]}\right|_{x_1}^{x_2} + b$$

Non-dimensionalized with respect to the length (2b) this yields:

$$\frac{c.g._{shell}}{L} = \frac{1}{2}\,\frac{\left\{\left(\frac{\rho}{L}\right)^2 + 2\left[1-\left(\frac{\rho}{L}\right)^2\right]X^* - \left[1-\left(\frac{\rho}{L}\right)^2\right]X^{*2}\right\}^{3/2}}{3\left[\left(\frac{a}{b}\right)^2-1\right]\cdot\left[\frac{S_L}{2\pi D}\right]}\Bigg|_{X_1^*}^{X_L^*} + \frac{1}{2}$$

Weight Distribution

The weight of the intact vehicle, capsule, or remaining vehicle is composed of the weights of the individual components, the shell weight, and the weight of the closure plate. The weights of the individual components are assumed to be known. The weight of the ellipsoidal shell is assumed to be the surface area of the shell times the appropriate thickness times the specific weight of the appropriate material. The closure plates are assumed to be flat plates of the same thickness and material as the capsule or remaining vehicle shell. The weight of the closure plate is then the area of the plate timesthe thickness times the specific weight of the appropriate material.

Mathematically,

$$W_{total} = \sum_i W_i + W_{shell} + W_{plate}$$

where the W_i are the individual component weights.

The longitudinal center of gravity is given by the summation of weights times the moment arm, divided by the total weight:

$$\overline{LCG} = \frac{\sum W_i x_i + W_{shell} X_{cgshell} + W_{plate} X_{sep}}{W_{total}}$$

Similarly, the vertical or radial center of gravity:

$$\overline{VCG} = \frac{\sum W_i y_i}{W_{total}}$$

It should be noted that there is no contribution to the vertical c.g. from either the shell or the closure plate.

The subroutine WGTS calculates the weights of the intact vehicle, the capsule, and the remaining hull, and their respective logitudinal and vertical centers of gravity.

V. HYDRODYNAMIC COEFFICIENTS

Accurate prediction of the hydrodynamic coefficients is a requirement for an accurate dynamic simulation of the motions of any underwater vehicle. It is unfortunate that there is such a paucity of experimental data and/or theoretical results which can be used in studies of the type undertaken here. The vehicles which are being modelled are truncated ellipsoids of revolution (see Section IV) with L/D ratios between 1 and 10. We assume that the intact submarine is an ellipsoid of revolution and thus both the separated capsule and the remaining vehicle are truncated ellipsoids. Both the capsule and the remaining vehicle may operate at very large angles of attack, from 0 to 360 degrees.

Because of the lack of hydrodynamic data on truncated ellipsoids of revolution, considerable engineering judgement has been necessary in order to formulate the equations of motion. Where good data was available, it was used. Where the data is questionable, simplified estimates were used.

V-1 The Lift and Drag Coefficients

The resistance of the body may be thought of as composed of three parts:

a. Skin Friction
b. Pressure Drag
c. Base Drag

Assuming that the flow is turbulent the two-dimensional skin friction coefficient is given by

$$C_f = \left[\frac{1}{3.46 \ln(R_e) - 5.6}\right]^{1/2}$$

where $C_f = \dfrac{D_f}{\frac{1}{2}\rho V^2 S}$, ln denotes the natural logarithm, and $R_e = \dfrac{V\ell}{\nu}$ is the Reynold's number based on the length of the body. Skin friction coefficients for three-dimensional bodies are larger because the 3-D relief effect thins the boundary layer and hence increases the shear stress at the wall. For turbulent flow the increase in C_f due to three-dimensional effects can be written (See Reference 1)

$$\Delta C_f = 0.0016 \; \ell/d \, [R_e]^{-2/5}$$

The pressure drag for ellipsoidal bodie operating at Reynolds numbers above 7.5 X 10^5 can be xpressed as a polynomial in d/ℓ (Reference 1)

$$C_{d_0} = 0.0875714 - 0.119702\left(\frac{d}{\ell}\right) + 0.133631\left(\frac{d}{\ell}\right)^2$$

If we define $C_{f_B} = C_f \frac{S}{S_B}$ where S_B is the base area, then the base drag for 3-D bodies can be written (Reference 1)

$$C_{DB} = \frac{0.029}{\sqrt{C_{f_B}}}$$

For bodies with a base diameter smaller than the diameter of the maximum cross-section we have

$$C_{DB} = \frac{0.029\left(\frac{d_B}{d_{max}}\right)^3}{\sqrt{C_{f_{max}}}}$$

where

$$C_{f_{max}} = C_f \frac{4S}{\pi d_{max}^2}$$

The total drag at zero angle of attack is then

$$D = C_D \, \tfrac{1}{2} \rho V^2 S$$

where

$$C_D = C_f + \Delta C_f + C_{d_0} + C_{DB} \frac{S_B}{S}$$

The lift coefficient may initially be approximated by the cross flow principle for circular cylinders in laminar flow (see Figure V-1):

$$C_L = C_s \sin^2\alpha \cos\alpha \frac{S_P}{S}$$

$$C_{D1} = C_s \sin^3\alpha \frac{S_P}{S}$$

where C_S is the side force coefficient based on the projected area

$$C_s = \frac{\text{side force}}{\frac{1}{2}\rho V^2 S_P} \approx 0.2$$

$$C_L = \frac{L}{\frac{1}{2}\rho V^2 S_p}$$

$$C_{D1} = \frac{D1}{\frac{1}{2}\rho V^2 S_p}$$

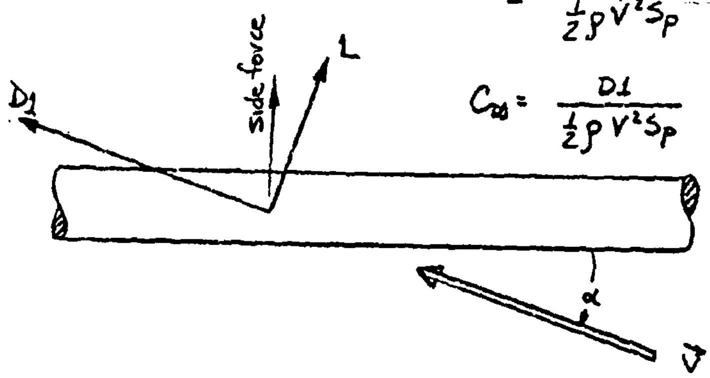


Figure V-1

We can now write a relationship for the lift and drag of the capsule which depends on the length-diameter ratio and the angle of attack:

$$C_L = C_S\,(\sin^2\alpha \cos\alpha)$$

$$C_D = C_{D1} + C_D \cos^2\alpha$$

The above lift and drag coefficients are admittedly rough approximations but better estimates would require that a comprehensive experimental program be undertaken.

V-2 Added Mass Coefficients

When either the escape capsule or the remaining vehicle is subjected to external forces and/or moments (buoyancy, hydrodynamic, etc.) not only must the mass of the vehicle itself be accelerated but also the mass of some of the surrounding water. Newton's second law appears as

$$\vec{F} = (m + m_v)\,\vec{a}$$

for linear motions;

$$\vec{M} = (I + I_v)\,\ddot{\theta}$$

for angular motions; where m_v and I_v are termed the virtual moment of inertia, respectively. In the notation of Section III, Newton's second law becomes for motion in the x, y plane

$$(M + M_x)\,\ddot{x} = \vec{F}_x$$
$$(M + M_y)\,\ddot{y} = \vec{F}_y$$

where M_x and M_y are the virtual masses in the x and y directions. The virtual mass is expected to be proprotional to the displaced mass of fluid

$$m_{vx} = \rho \nabla k_x \quad , \quad m_{vy} = \rho \nabla k_y$$

where K_x and K_y are the virtual mass coefficients. Similarly, the virtual moment of inertia is assumed to be proprotional to the moment of inertia of the displaced mass about an axis through the center of buoyancy; hence

$$I_v = \rho \nabla r_g^2 k_\theta$$

where r_g is the radius of gyration of the displaced volume and k_θ is the virtual moment of inertia coefficient.

K_θ will not depend on the body orientation, but K_x and K_y definitely will. We will assume in this study that K_x and K_y can be written

$$k_x = K_a |\cos\alpha| + K_c |\sin\alpha|$$

$$k_y = K_a |\sin\alpha| + K_c |\cos\alpha|$$

where K_a and K_c are the axial and cross-flow added mass coefficients.

To the best of our knowledge, values of K_a and K_c for truncated ellipsoids of revolution are not available either in the experimental or theoretical literature. Even if they were, the results would be of questionable value in this work because the motions of the capsule are, for the most part, nonoscillatory and nearly all of the published data is for small amplitude, oscillatory motions where potential flow can be assumed. The virtual mass coefficient of a blunt body when the flow is separated is much harder to analyze and measure because it not only depends on the shape of the body but also on the velocity and acceleration in a complicated way. Also, because of viscous effects, the virtual mass not only depends on the instantaneous acceleration and velocity but also on the motion-time history. Despite the difficulties in estimating the true virtual masses it appears that the potential flow values do give reasonable results, at least for the initial acceleration from rest. However, even potential flow virtual mass coefficients are not available in the literature for truncated ellipsoids of revolution. Lamb (Reference 2) derives the K_x, K_y, and K_θ for intact ellipsoids of revolution. These results follow:

$$k_a = \frac{\alpha_0}{2-\alpha_0}$$

$$k_c = \frac{\beta_0}{2-\beta_0}$$

$$k_\theta = \frac{e^4(\beta_0-\alpha_0)}{(2-e^2)[2e^2-(2-e^2)(\beta_0-\alpha_0)]}$$

where

$$e = \text{eccentricity} = \frac{\sqrt{b^2-a^2}}{b} = \sqrt{1-\left(\frac{D}{L}\right)^2}$$

$$\alpha_0 = \frac{2(1-e^2)}{e^3}\left[\frac{1}{2}\ln\left(\frac{1+e}{1-e}\right)-e\right]$$

$$\beta_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3}\ln\left(\frac{1+e}{1-e}\right)$$

and a and b are the semi-minor and semi-major axes of the ellispoid, respectively. As an initial approximation, the virtual mass coefficients of a truncated ellipsoid are assumed to be the same as the coefficients of a whole ellipsoid of revolution with the same L/D ratio.

This assumption, of course, neglects the effect of the truncation, but hopefully includes the effect of L/D on the virtual mass coefficients. Figure V-2 shows how K_a, K_c, and K_θ vary with L/D.

V-3 Hydrostatic Parameters

The motion of the capsule will be influenced to a large extent by the buoyant forces and moments. The buoyant force, B, acts upward through the centroid of the displaced volume, b. The weight, W, acts downward through the center of gravity, g. The magnitude of the ratio

$$C_1 = \frac{W}{B} = \frac{W}{\rho\nabla}$$

determines whether the capsule will rise or sink. If $C_1 < 1$ the capsule will rise. If $C_1 > 1$ the capsule will sink. If $C_1 = 1$ the capsule is neutrally buoyant and will not move vertically except under the influence of hydrodynamic forces. The hydrostatic moment

$$M_s = B\left[(d_{g_1}-d_b)\cos\theta + d_{g_2}\sin\theta\right]$$

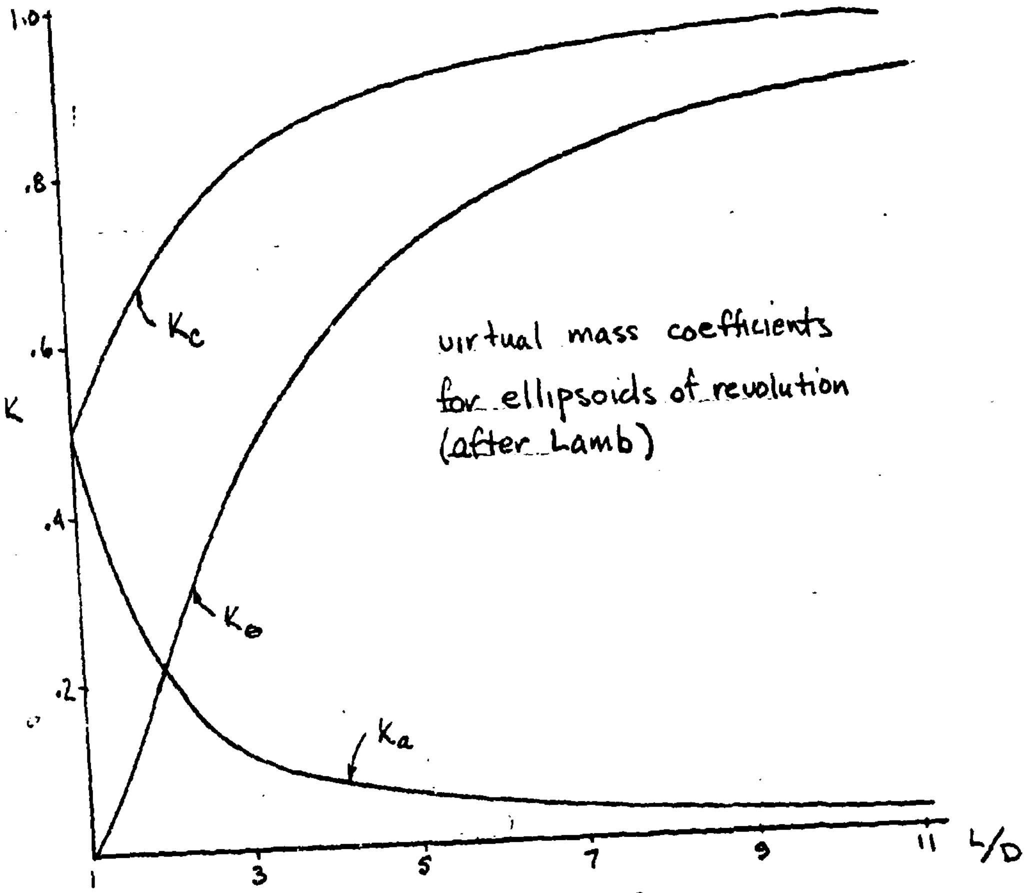


Figure V-2

is a function of θ and is zero if g and b are colinear vertically. This angle denoted by θ_o (See Figure . V-3),

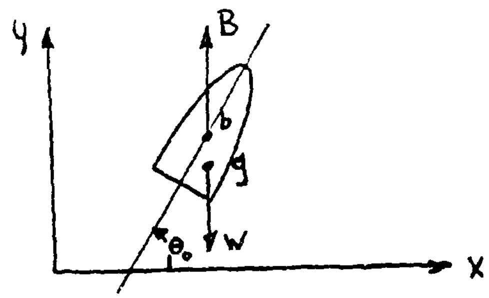
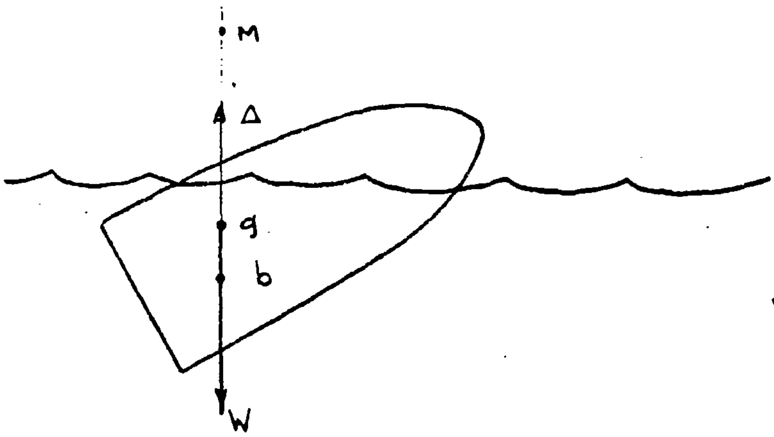


Figure V-3

is the angle which the capsule would assume if the hydrodynamic forces were zero and it was free to trim. The capsule should assume an angle close to θ_o after the angular oscillations have damped out and the capsule is in steady motion long after it has separated from the submarine.

VI. SIMULATION OF THE MOTIONS OF THE CAPSULE ON THE SEA SURFACE

The attitude of the capsule when it reaches the surface will, together with the sea state, determine the magnitude of the heave and pitch motions experienced by the capsule. The first task is to determine the attitude the capsule will assume when floating on the surface waiting for rescue. It will rise until the displacement is just equal to the weight and will trim until the centroid of the displaced volume is vertically colinear with the center of gravity and the metacentric height is positive (See Figure VI-1). Since there may be more than one stable attitude, we will assume that eventually the capsule will attain the most stable orientation, that is, the orientation which yields the largest metacentric height.



Δ = displacement
W = weight
g = center of gravity
b = center of buoyancy
M = metacenter

Figure VI-1. Capsule Floating on the Surface

Having reached equilibrium on the surface, it will be excited in heave and pitch by the waves. In order to simplify the difficult problem of predicting the heave and pitch for arbitrary capsule attitudes, several assumptions will be made:

a. The capsule is assumed to be slender enough so that it will be stable in one of three orientations:

1. vertical, bow up
2. vertical, bow down
3. nearly horizontal

In other words we are assuming that the capsule will behave similarly to a spar buoy.

b. If the capsule is most stable vertically we assume that the pitch angle is always zero and only heaving motion is significant.

c. If the capsule is most stable horizontally we assume that the pitch angle is equal to the wave slope and the heave amplitude is equal to the wave amplitude.

With the above assumptions the problem becomes one of determining the most stable attitude of the three possible, and then calculating heave and pitch amplitude spectrums from the specified sea state.

The Most Stable Attitude

We imagine the capsule to be enclosed in a rectangular box as in Figure VI-2.

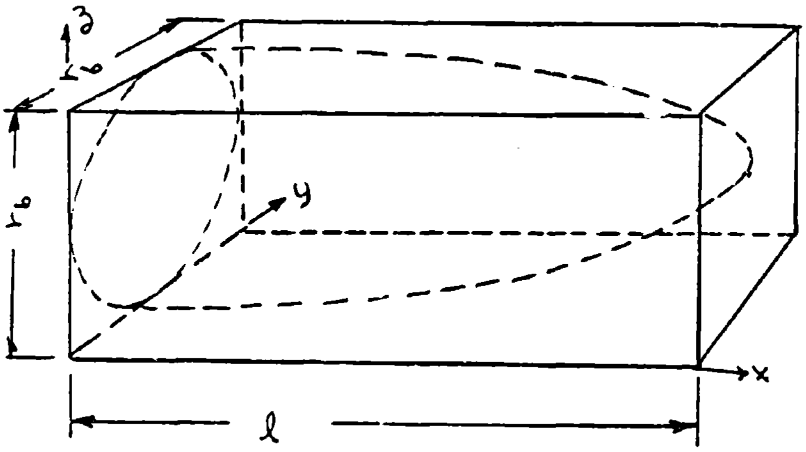


Figure VI-2. Digital model of capsule.

The box is now divided into 10^6 small rectangular cells, each having dimensions $\ell/100 \times r_b/100 \times r_b/100$. Wherever the capsule occupies any part of a cell it is considered filled with material. We now have a representation of the capsule volume which is not exact but is sufficiently accurate for our purposes and is very simple to manipulate in the computer.

The capsule can be represented by a three-dimensional matrix V(1,1,1) ... V(100,100,100) the elements of which are 1 if the capsule occupies the cell and 0 otherwise. The matrix is arranged so that the first index represents locations on the x axis, the second index represents the y axis and the third the z axis. For example, V(1, 50, 50) is the cell at the center of the separation plane; V(100, 50, 50) is the cell at the nose of the capsule.

The equilibrium draft of the capsule and other hydrostatic parameters necessary to compute the metacentric height at each of the three allowable attitudes are easy to calculate once the digital capsule model has been generated.

The draft is found by counting the occupied cells starting at a base plane which is parallel to the waterplane. Occupied cells are counted layer by layer until the number of cells multiplied by a scale factor just exceeds the weight of the capsule. The waterplane is then located at the layer of cells at which the displacement most nearly equals the weight. At the same time the equilibrium drafts are found, the first moments of the three underwater volumes about the baseplane parallel to the waterplane can be calculated by multiplying each element by its distance from the appropriate baseplane.

The drafts for the three attitudes--horizontal (d_h), vertical bow down (d_{v1}), vertical bow up (d_{v2}) can be written

$$d_h = r_b \frac{n_b}{100}$$

$$d_{v1} = \ell \frac{100 - n_{v1}}{100}$$

$$d_{v2} = \ell \frac{n_{v2}}{100}$$

where n_{v1}, n_h, and n_{v2} are found from the following expressions:

$$\upsilon = \sum_{k=1}^{n_h} \sum_{j=1}^{100} \sum_{i=1}^{100} V(i,j,k)$$

$$\upsilon = \sum_{i=100,-1}^{n_{v1}} \sum_{j=1}^{100} \sum_{k=1}^{100} V(i,j,k)$$

$$\upsilon = \sum_{i=1}^{n_{v2}} \sum_{j=1}^{100} \sum_{k=1}^{100} V(i,j,k)$$

where

$$\upsilon = \frac{W}{4\gamma r_h^2 \ell} \times 10^6$$

The vertical moments M_h, M_{v1}, and M_{v2} can be written in a similar fashion once n_h, n_{v1}, and n_{v2} are known:

$$M_h = \frac{8\gamma r_h^3 \ell}{10^8} \sum_{k=1}^{n_h} \sum_{j=1}^{100} \sum_{i=1}^{100} k\,V(i,j,k)$$

$$M_{v1} \quad \frac{4\gamma r_h^2 \ell^2}{10^8} \sum_{i=100,-1}^{n_{v1}} \sum_{j=1}^{100} \sum_{k=1}^{100} (100-i)\; V(i,j,k)$$

$$M_{v2} \quad \frac{4\gamma r_h^2 \ell^2}{10^8} \sum_{i=1}^{n_{v2}} \sum_{j=1}^{100} \sum_{k=1}^{100} i\; V(i,j,k)$$

The heights of the centroid from the appropriate base-plane can now be computed:

$$b_h = \frac{M_h}{W}$$

$$b_{v1} = \frac{M_{v1}}{W}$$

$$b_{v2} = \frac{M_{v2}}{W}$$

The metacentric radii of each waterplane can be written

$$\overline{BM}_h = \frac{2r_b}{100}\left[\frac{\sum_{i=1}^{100}\sum_{j=1}^{100}(j-50)^2\, v(i,j,n_h)}{\upsilon}\right]$$

$$\overline{BM}_{v1} = \frac{4r_b^2}{100\ell}\left[\frac{\sum_{j=1}^{100}\sum_{k=1}^{100}(j-50)^2\, v(n_{v1},j,k)}{\upsilon}\right]$$

$$\overline{BM}_{v2} = \frac{4r_b^2}{100\ell}\left[\frac{\sum_{j=1}^{100}\sum_{k=1}^{100}(j-50)^2\, v(n_{v2},j,k)}{\upsilon}\right]$$

And, finally, the metacentric height for the capsule floating at each of the three attitudes can be written:

$$\overline{GM}_h = \overline{BM}_h + b_h - g_h$$

$$\overline{GM}_{v1} = \overline{BM}_{v1} + b_{v1} - g_{v1}$$

$$GM_{v2} = \overline{BM}_{v2} + b_{v2} - g_{v2}$$

where g_h, g_{v1}, g_{v2} are the distances of the center of gravity from the appropriate base line planes.

The most stable attitude and hence the most probable attitude is the attitude which yields the largest $\overline{GM}$.

The Pierson-Moskowitz Spectrum is widely accepted by naval architects since its introduction about 1965 as a realistic representation of fully developed, long-crested, deep-water, irregular waves. The spectrum is based on the analysis of several hundred sea wave records recorded by British scientists. Mathematically, the spectrum can be written

$$S(\omega) = \frac{A}{\omega^5}\, e^{-B/\omega^4}$$

where

$S(\omega)$	=	Spectrum ordinate (ft^2-sec)
ω	=	Circular wave frequency (rad/sec)
A	=	$.0081\ g^2$
B	=	$33.56/(\bar{H}_{1/3})^4$
g	=	Acceleration of gravity
$\bar{H}_{1/3}$	=	Significant wave height $= 4\sqrt{E_s}$
E_s	=	$\int_0^\infty S(\omega)\, d\omega \propto$ energy in the sea state

To obtain the spectrum as a function of the sea state, we use the following tabular relationship between $\bar{H}_{1/3}$ and the sea state:

Sea State	$\bar{H}_{1/3}$ (ft)
0	0.10-0.15
1	0.50-1.20
2	1.50-3.00
3	3.50-5.00
4	6.00-7.50
5	8.00-12.00
6	14.00-20.00
7	25.00-40.00
8	45.00-60.00
9	70.00-100.00

The maximum wave slope of deep water waves can be written

$$\theta_{max} = \frac{2\pi\zeta}{\lambda}$$

where ζ is the wave amplitude. But for deep water waves, $\lambda = 2\pi g/\omega^2$. Therefore $\theta_{max} = \omega^2\zeta/g$ and the ordinates of the slope spectrum can be obtained from the amplitude spectrum:

$$S'(\omega) = \frac{2\omega^4}{g^2} S(\omega)$$

We are now ready to calculate the pitch and heave response of the capsule, since the attitude is known. If the capsule is stable in the upright attitude then the pitch angle can be assumed to be zero and the heave amplitude can be written

$$A_h^2(\omega) = \frac{(c - a\omega^2)^2 + b^2\omega^2}{(c - m\omega^2 - a\omega^2)^2 + b^2\omega^2}\, e^{-\frac{2\omega^2}{g}d}\, S(\omega)$$

where

C = ρg A waterplane
a = added mass $\cong$ .5m
b = damping coefficient
m = mass of capsule
d = draft of capsule

If the capsule is most stable in the horizontal attitude, the pitch angle is assumed to be equal to the wave slope and the heave amplitude is assumed to be in phase and equal to the wave amplitude. With these assumptions we can write

$$\theta^2(\omega) = S'(\omega)$$

$$A_h^2(\omega) = S(\omega)$$

and

$$\bar{\theta}_{1/3} = 4.0 \int_0^{\infty} \theta^2(\omega)\, d\omega$$

$$\bar{A}_{1/3} = 4.0 \int_0^{\infty} A^2_h(\omega)\, d\omega \ .$$

VII. THE DYNAMIC SIMULATION

One of the primary objectives of this study is to develop a dynamic simulation of a modular submarine which would allow the researcher to study in a realistic fashion the effects of changes in the physical configuration of a modular submarine on the dynamics of the separation and ascent of the forward capsule.

The simulation was required to have several important features:

(a) The simulation must be capable of handling arbitrary vehicle weight distributions.

(b) It must be able to handle various vehicle geometries.

(c) Various types of casualties must be able to be simulated.

(d) It must be user-oriented.

(e) It must operate in as close to a real-time environment as possible.

To satisfy the above requirements a digital simulation model was developed for a time-shared PDP-10 computer. Input and output is provided via a Typagraph interactive terminal. The simulation model takes advantage of the Typagraph's graphical capability and provides graphical output of either the capsule or the remaining vehicle motion after separation occurs.

The simulation model appears to the user as simply a list of commands which he types on the terminal. When a command is typed the computer interprets the command and performs certain calculations or input/output. After a command has been executed the computer types two asterisks (**) to indicate that it is ready for another command. The simulation is started by typing

RUN DSK:MODSEC⟩

Note: ⟩ is a carriage return and is typed after each line.

The following section describes the simulation commands. They are listed in the approximate order that they might be used in a sample simulation (see Section VII for some sample runs). Each command may be abbreviated by the <u>underlined</u> letters.

Command	Description
**HELP}	Prints the contents of the data file HELP.DAT which contains a list of commands with brief descriptions as well as hints for running the simulation. Also included in the HELP.DAT file is a list of modifications.
**INPUT XXXX}	The input commands are two-word commands used to input data for a simulation. The two words must be separated by a blank and each word can usually be abbreviated by its first letter. All input commands may be repeated individually during a simulation. This feature gives the user a convenient method of updating or changing single input parameters without having to enter all of the variables each time.
**INPUT GEOMETRY}	The computer asks for the L/D ratio, overall length, and location of the separation bulkhead. The L/D ratio is the L/D of the intact submarine, the length is in feet, and the location of the separation bulkhead is given as a decimal fraction of the overall length. That is, a number between 0 and 1.0.
**INPUT WEIGHTS}	Allows the user to either type in a new weight distribution consisting of a series of component weights, together with the horizontal and vertical locations of each weight or retrieve a previously entered weight distribution from a disc file.
**INPUT VOLUMES}	Allows the user to type in a list of floodable spaces which will be used later to simulate a casualty. Casualties are simulated by flooding user-specified volumes. As with the weight distribution the option of retrieving a previously stored volume distribution from a disc file is provided.

**UPDATE) This command gives the user the capability to update, change, and add to the current weight or volume distribution. This command prompts the computer to ask a series of questions regarding the exact nature of the updating to be done; these questions are self explanatory. See the example problems.

**INPUT DEPTH) Allows the user to type in the depth at which the casualty and separation is to occur. This number is not used in any calculations.

**INPUT SPEED) Allows the user to type in the initial horizontal velocity and vertical velocity in ft/sec. This is the speed of the submarine at the instant of separation.

**INPUT TRIM) Allows the user to type in the trim angle of the submarine at the instant of separation. The command TRIM should be used to guide the user in his choice of initial trim angle.

**WEIGHTS) Calculates the various vehicles' weights and the longitudinal and vertical centers of gravity. This command is given following INPUT WEIGHTS or in the case of a damaged vehicle DAMAGE, but always prior to ASCEND or HULL.

**INPUT SEAS) Allows the user to type in the surface sea state as an integer from 0 to 9. This value is used to calculate a sea spectrum which in turn is used to calculate the pitch and heave response of the capsule on the surface.

**DAMAGE) This command allows the user to specify which of the spaces he wishes to flood. This command simulates various casualty types by flooding the specified spaces.

**SEPARATE) Separates the capsule from the submarine and initializes parameters for the ascent phase. This command does not produce any output or require any input.

**TRIM)	Calculates the net buoyant force on the intact vehicle and the trim angle the intact vehicle would take if it were allowed to reach equilibrium. This command uses the intact vehicle in its current condition (either damaged or undamaged) for the calculations and prints out the results on the terminal.
**ASCEND)	Calculates the trajectory, angular orientation, horizontal velocity of the capsule as a function of time. t=0 is assumed to be the time that separation occurs. Before the calculation is done the user must input the maximum time in seconds he wishes to calculate data.
**HULL)	Essentially the same as "ASCEND" except that the trajectory of the remaining hull is calculated.
**GRAPH)	Generates an annotated plot on a Typagraph Model 3 terminal of the horizontal and vertical distance from the separation point and the angular orientation of the vehicle as a function of time. This command plots the most recently generated trajectory. For example, if "HULL" was typed just prior to "GRAPH" the data plotted would be for the remaining vehicle. If "ASCEND" was typed immediately prior to "GRAPH" the data plotted would be for the capsule.
**MOTIONS)	Calculates and plots the behavior of the capsule at the surface. Both the pitch and heave response spectrums are plotted together with the capsule orientation, the average wave height, heave amplitude and pitch angle.
**PRINT)	Prints, in tabular form, the dimensional and non-dimensional results of the last calculated trajectory. This command is used primarily when running the simulation from an ordinary Model 33 teletype. Before printing the data the computer asks how many points the user wishes to print out. The maximum number of points you can print is 101.

**CHECK}   Prints the current status of the simulation (what commands have been executed up to the present time) and all current input parameters. The user has the option of printing out the rather lengthy weight and volume distributions.

**NEW CASE}   Initializes the system and prepares it for a new simulation. All previous data except stored weight and volume distributions will be lost to the user.

VIII. SAMPLE SIMULATIONS

A. Illustrative example

In order to familiarize the user with MODSEC system, a short, simple example will be set up and executed.

The first step in using the simulation system is to choose a length/diameter ratio, and overall length of the submarine:

L/D = 10.0
Length = 300.0 ft.

The position of the separation plane is selected next. This is one of the variables that the user will probably wish to vary during the simulation to see its effect on the capsule motion. Initally let

$$x_o = 0.4$$

the weight distribution is the next and most difficult parameter to select. The more accurate and detailed the weight distribution is, the more meaningful the simulation will be. Ideally, the user should perform a detailed weight analysis on the proposed concept and generate a list of components, their weights, and locations in the submarine. This list is then used as input data to the simulation system. For the first example, we will assume that the weight distribution is as given in the following table. This distribution is approximately parabolic over the length and yields a total submarine weight of 4,046 long tons.

Weight	Long'l Cg	Vertical Cg
56	.05	0
108	.10	.15
155	.15	-.30
192	.20	.20
275	.25	-.05
254	.30	0
271	.35	-.5
288	.40	-.45
298	.45	-.20
301	.50	-.45
297	.55	-.9
289	.60	-.8
272	.65	.2
255	.70	.35
225	.75	-.45
192	.80	-.20
154	.85	.10
108	.90	0
56	.95	0

The vertical centers are selected at random and vary from -1.0 to 1.0 depending on what portion of the local radius each weight is from the centerline.

The thickness of the capsule shell is assumed to be 2.0 inches. The thickness of the main hull is assumed to be 1.7 inches. Both shells are assumed to be constructed of steel with a density of 487 lbs/ft^3.

The size and location of the spaces which are likely to be flooded in the event of a casualty are computed next. In this simple example, we assume that there are only two floodable spaces:

Volume	Long'l Center	Vertical Center
8000 ft^3	.70	.1
2000 ft^3	.20	-.3

We now have enough data to begin the simulation of the modular submarine shown in Figure VIII-1.

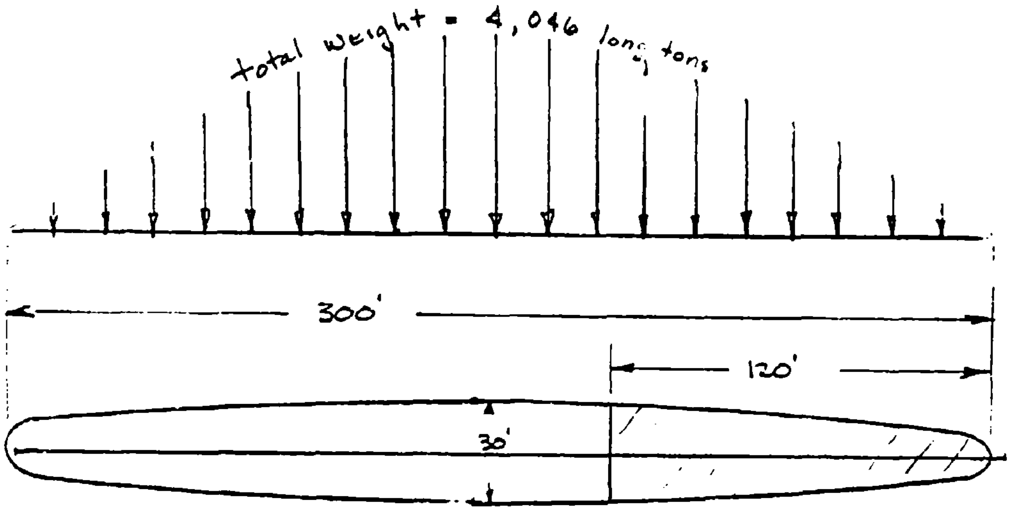


Figure VIII-1. Modular submarine, Example 1.

We begin by dialing the telephone number of the computer. The user dialog then proceeds as follows: (NOTE: User responses are underlined. The RETURN key must be depressed after each line of user response. In the documentation, the carraige return key is denoted by ⤶)

ON-LINE SYSTEMS 16 1324 9-20-70 *BENCHMARK SERVICE	The computer prints a heading.
USER NUMBER?..99,88⤶	The user supplies his assigned number. The comma is necessary.
[redacted]⤶	A mask is printed by the system. The user enters his assigned password over the mask.
INFORM UPDATED AUG 4	If the user responds INFORM to SYSTEM?.., information will be printed at his terminal.
SYSTEM?..	The time-sharing system is ready to accept commands when a period (.) is printed. If an error occurs during the simulation, a period will be printed. Type ST⤶ to continue the simulation. The command RUN DSK:MODSEC⤶ loads the MODSEC System from a disc file and begins execution.

```
.RUN DSK:MODSEC.SAV

IF YOU REQUIRE INSTRUCTIONS, TYPE "HELP".
**HELP



** ABBREVIATED USER INSTRUCTIONS:


THE PROGRAM YOU ARE RUNNING WILL HELP YOU DESIGN AN ELLIPSOIDAL SUB-
MARINE WITH A SEPARABLE CAPSULE. EACH COMMAND BELOW EITHER PERFORMS A
COMPLETE FUNCTION, PROMPTS ADDITIONAL QUESTIONS FROM THE PROGRAM,
OR PROVIDES YOU WITH INFORMATION. THE PROGRAM SIGNALS THAT IT IS
READY FOR A COMMAND BY PRINTING  **  AT THE LEFT MARGIN OR FOR A RE-
SPONSE BY PRINTING  --  .
```

COMMAND	DESCRIPTION
HELP	PRINTS USER INSTRUCTIONS
INPUT GEOMETRY	LETS USER SPECIFY GENERAL GEOMETRIC CHARACTERISTICS OF VEHICLE
INPUT WEIGHTS	LETS USER SPECIFY WEIGHT DISTRIBUTION OF VEHICLE
INPUT VOLUMES	LETS USER SPECIFY FLOODABLE VOLUME DISTRIBUTION
INPUT DEPTH	LETS USER SPECIFY INITIAL DEPTH
INPUT SPEED	LETS USER SPECIFY INITIAL VERTICAL AND HORIZONTAL VELOCITIES
INPUT TRIM	LETS USER SPECIFY INITIAL TRIM ANGLE
INPUT SEAS	LETS USER SPECIFY SURFACE SEA STATE
UPDATE	ALLOWS CHANGES TO BE MADE TO WEIGHT OR VOLUME DISTRIBUTIONS
WEIGHTS	CALCULATES WEIGHTS AND CENTERS
DAMAGE	SIMULATES SELECTIVE FLOODING OF VOLUMES
TRIM	CALCULATES BOUYANT FORCE AND TRIM OF INTACT VEHICLE
SEPARATE	SIMULATES CAPSULE SEPARATION FROM HULL
ASCEND	CALCULATES CAPSULE TRAJECTORY
HULL	CALCULATES REMAINING HULL TRAJECTORY

GRAPH	PLOTS BEHAVIOR OF SEPARATED CAPSULE OR REMAINING VEHICLE (WHICHEVER HAD THE MORE RECENT TRAJECTORY COMPUTED)
MOTION	DETERMINES THE ATTITUDE OF CAPSULE ON THE SURFACE AND PLOTS THE SEA SURFACE SPECTRUM
PRINT	PRINTS TRAJECTORY DATA IN TABULAR FORM
CHECK	PRINTS CURRENT STATUS OF SIMULATION
NEW CASE	INITIALIZES PROGRAM FOR A NEW SIMULATION

FOR ADDITIONAL DETAILS, CONSULT PROGRAM DOCUMENTATION

**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 10,300,.4

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 2.0,1.7

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) -- 487.0,487.0

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO --- old weights might have been stored on a disc file

HOW MANY WEIGHTS ARE TO BE INPUT? -- 19

INPUT WEIGHTS IN LONG TONS, LONG'L CG, VERT CG
IN GROUPS OF THREE

56,.05,0 --56 ton weight, .05L from the bow, on the submarine centerline

108,.10,.15

155,.15,-.30

192,.20,.20

275,.25,-.05

254,.30,0

271,.35,.5

288,.40,.45

298,.45,-.20

301,.50,-.45

297,.55,-.9

289,.60,.8

272,.65,.2

255,.70,.35

225,.75,.45

192,.80,-.20

154,.85,.10

108,.9,0.

56,.95,.0

STORE WEIGHTS? (YES OR NO) -- NO --weights may be stored permanently on a disc file

**UPDATE --used to change weight or volume array

```
DO YOU WISH TO UPDATE WEIGHTS OR VOLUMES?
TYPE V OR W -- W

IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 192,.20,.20
```

--- weight to be changed

```
INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 192,.20,-.20
```

---and replaced by this new one

```
DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- Y
```

changing vertical centers of selected weights so the CG will be below the center-line.

```
IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 271,.35,.5

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 271,.35,-.5

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- Y

IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 288,.4,.45

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 288,.4,-.45

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- Y

IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 289,.6,.8

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 289,.6,-.8

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- Y

IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 225,.75,.45

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 225,.75,-.45

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- Y
```

```
IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 192,.8,-.20

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 195,.8,-.9

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- N

DO YOU WISH TO STORE THIS DISTRIBUTION FOR FUTURE USE

TYPE Y OR N -- Y                        ---we now store the weights

ENTER SHIP "IDENTIFIER", (5 CHARACTERS) -- SUB01 ---under this name

DISTRIBUTION HAS BEEN STORED FOR FUTURE USE

**WEIGHTS    --- calculates total weights and centers.

**INPUT VOLUMSΔSΔES  ---Δ means that a "rubout" was typed to delete the
                        character just typed. The computer then typed the
DO YOU WANT TO USE OLD FLOODABLE VOLUMES?   deleted character and another Δ
YES OR NO -- NO

HOW MANY FLOODABLE VOLUMES -- 2

INPUT FLOODABLE VOLUMES, LONG'L, AND VERTICAL CENTROIDS
IN GROUPS OF THREE
8000,.70,.1 ---8000 cu. ft. of space is located .7L from the bow and .1R
2000,.20,-.3                      above the centerline.

STORE VOLUMES? (YES OR NO) -- YES   ---floodable spaces are stored on a disc
                                       file
ENTER SHIP "IDENTIFIER", (5 CHARACTERS) -- SUB01 ---under this name

DISTRIBUTION HAS BEEN STORED FOR FUTURE USE

**TRIM --calculates the attitude of intact vehicle. Used to balance
                                                          submarine

INTACT VEHICLE IS HEAVY OVERALL BY   1764619. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN   23.45 DEGREES
**INPUT GEOMETRY                                   changing the
                                                   L/D ratio to get
SPECIFY L/D, LENGTH, SEPARATION PLANE -- 8.8,300,.4  a neutrally
                                                   buoyant submarine
**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY    871201. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN   23.45 DEGREES
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 8.85,300,.4
```

```
**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY    739556. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN   23.45 DEGREES
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 8.85,300,.4

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY    495858. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN   22.47 DEGREES
**UPDATE

DO YOU WISH TO UPDATE WEIGHTS OR VOLUMES?
TYPE V OR W  -- W

IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C

INPUT PRESENT WEIGHT, LCG, AND VCG -- 56,.95,0

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 100,.95,0

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- N

DO YOU WISH TO STORE THIS DISTRIBUTION FOR FUTURE USE

TYPE Y OR N -- Y

ENTER SHIP "IDENTIFIER", (5 CHARACTERS)  -- SUB01

IS DISTRIBUTION NEW OR AN UPDATE ?
TYPE N OR U -- U

DISTRIBUTION HAS BEEN STORED FOR FUTURE USE

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY    397298. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN    8.58 DEGREES
**UPDATE

DO YOU WISH TO UPDATE WEIGHTS OR VOLUMES?
TYPE V OR W  -- W

IS DISTRIBUTION TO BE UPDATED TO BE CHANGED, DELETED, OR ADDED?
TYPE C,D, OR A -- C
```

adding a weight to the stern to bring the submarine to an even keel attitude

INPUT PRESENT WEIGHT, LCG, AND VCG -- 100,.95,0

INPUT NEW WGT OR VOLUME, AND LCG, VCG -- 120,.95,0

DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?
TYPE Y OR N -- N

DO YOU WISH TO STORE THIS DISTRIBUTION FOR FUTURE USE

TYPE Y OR N -- Y

ENTER SHIP "IDENTIFIER", (5 CHARACTERS) -- SUB01 -modified weight distribution has been stored

IS DISTRIBUTION NEW OR AN UPDATE ?
TYPE N OR U -- U

IS DISTRIBUTION NEW OR AN UPDATE ?
TYPE N OR U -- U

DISTRIBUTION HAS BEEN STORED FOR FUTURE USE

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY 352498. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN 1.78 DEGREES
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 8.9,300,.4

$L/_D$ changed some more to make boat neutrally buoyant

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY 235069. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN 1.68 DEGREES
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 8.95,300,.4

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY 119358. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN 1.58 DEGREES
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 8.9,300,.4

**WEIGHTS

```
**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY    235069. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN    1.68 DEGREES
**INPU GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 9.0,300,.4

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY      6221. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN    1.49 DEGREES
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 9.01,300,.4 -- final L/D, Length,
                                                        Separation plane

**WEIGHTS

**TRIM

INTACT VEHICLE IS HEAVY OVERALL BY     16233. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN    1.47 DEGREES
**INPUT GEOMRTRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 9.0025,300,.4

**WEIGHTS

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY       600. POUNDS
INTACT VEHICLE IS TRIMMED BOW DOWN    1.48 DEGREES
**DAMAGE

DO YOU NEED A LIST OF FLOODABLE VOLUMES? (YES OR NO) -- YES

SPACE NO     VOL (CU.FT.)     LONG CG     VERT CG
--------     ------------     -------     -------

    1           8000.0         0.700       0.100
    2           2000.0         0.200      -0.300

HOW MANY SPACES DO YOU WANT TO FLOOD? -- 1      ---casualty sustained in
                                                   one space

INPUT THE NUMBERS OF FLOODED SPACES
1
                                     -----space no. 1 near the stern has
                                          been flooded
**WEIGHTS
```

**TRIM

DAMAGED INTACT VEHICLE IS HEAVY OVERALL BY 511400. POUNDS--intact vehicle will sink
DAMAGED INTACT VEHICLE WILL EVENTUALLY TRIM BOW UP 31.51 DEGREES
INTACT VEHICLE WILL TRIM HYDROSTATICALLY FROM -1.48
DEGREES BEFORE DAMAGE TO 31.51 DEGREES AFTER DAMAGE

**INPUT TRIM

SPECIFY INITIAL TRIM ANGLE (DEGREES) -- 31.51 ---same as hydrostatic trim angle (it need not be)

**INPUT SPEED

SPECIFY INITIAL VERTICAL AND HORIZONTAL SPEED (FT/SEC) -- -5.0,0.0--sinking slowly

**SEPARATE --capsule has separated from submarine

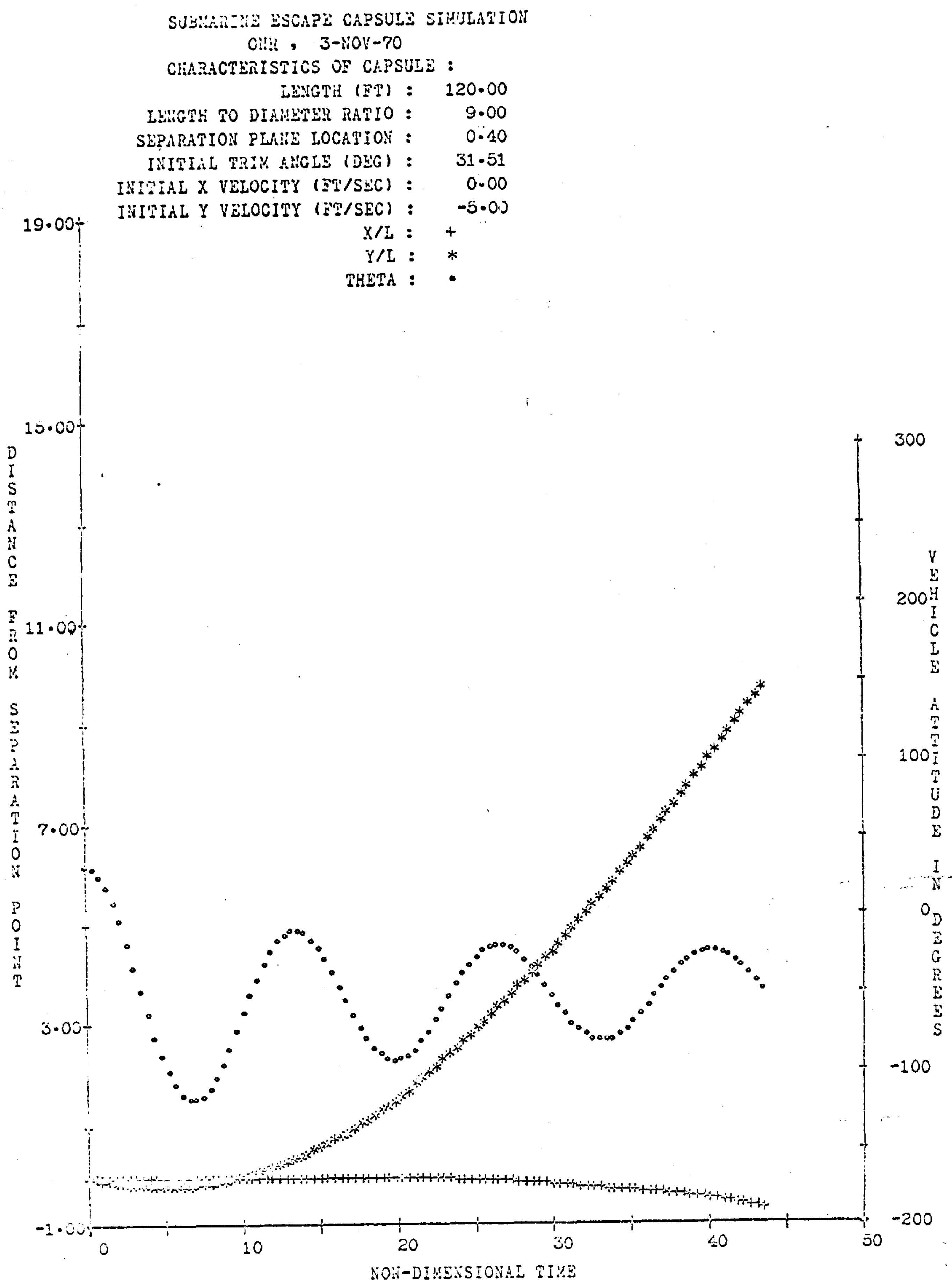
**ASCEND --trajectory is calculated

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 85 --- we are interested in the first 85 seconds after separation

**GRAPH -- we would like to see the results in graphical form

CHECK PLOT SWITCH --Typagraph plot switch moved to plot mode, carraige return typed.

Best Available Copy


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 120.00
LENGTH TO DIAMETER RATIO : 9.00
SEPARATION PLANE LOCATION : 0.40
INITIAL TRIM ANGLE (DEG) : 31.51
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : -5.00
X/L : +
Y/L : *
THETA : •
19.00
15.00
11.00
7.00
3.00
-1.00
DISTANCE FROM SEPARATION POINT
300
200
100
0
-100
-200
VEHICLE ATTITUDE IN DEGREES
0
10
20
30
40
50
NON-DIMENSIONAL TIME

**PRINT ---print tabular results of the last calculated trajectory,
in this case the capsule trajectory
HOW MANY POINTS DO YOU WISH TO PRINT (2 TO 14) -- 14

NON-DIMENSIONAL RESULTS

TIME	HORIZONTAL DISTANCE	HORIZONTAL VELOCITY	VERTICAL DISTANCE	VERTICAL VELOCITY	ATTITUDE DEGREES	ANGULAR VELOCITY
0.00	0.00	0.00	0.00	-0.08	31.51	0.00
3.08	-0.00	-0.00	-0.18	-0.04	-32.73	-35.07
6.16	-0.00	-0.00	-0.21	0.02	-114.38	-10.08
9.24	-0.00	-0.00	-0.08	0.07	-85.13	24.52
12.32	-0.00	-0.00	0.23	0.12	-17.22	12.07
15.40	-0.01	-0.00	0.66	0.16	-29.13	-17.68
18.48	-0.02	-0.00	1.21	0.20	-85.95	-11.98
21.57	-0.04	-0.00	1.93	0.26	-82.31	13.44
24.65	-0.05	-0.01	2.79	0.30	-32.72	12.71
27.73	-0.10	-0.02	3.73	0.31	-25.45	-8.34
30.81	-0.18	-0.03	4.73	0.34	-65.74	-12.78
33.89	-0.27	-0.03	5.84	0.37	-80.84	4.39
36.97	-0.35	-0.03	7.03	0.40	-48.19	12.96
40.05	-0.47	-0.05	8.28	0.41	-23.48	0.88
43.57	-0.68	-0.07	9.73	0.42	-43.27	-11.70

DIMENSIONALIZED RESULTS

TIME SECONDS	HORIZONTAL DISTANCE FEET	HORIZONTAL VELOCITY FT/SEC	VERTICAL DISTANCE FEET	VERTICAL VELOCITY FT/SEC	ATTITUDE DEGREES	ANGULAR VELOCITY DEG/SEC
0.00	0.00	0.00	0.00	-5.00	31.51	0.00
5.95	-0.09	-0.02	-21.78	-2.35	-32.73	-18.16
11.90	-0.18	-0.01	-25.77	1.15	-114.38	-5.22
17.85	-0.26	-0.01	-9.09	4.63	-85.13	12.69
23.80	-0.44	-0.07	27.87	7.53	-17.22	6.25
29.75	-1.13	-0.18	79.42	9.79	-29.13	-9.16
35.70	-2.65	-0.29	145.52	12.65	-85.96	-6.20
41.65	-4.32	-0.27	231.20	16.08	-82.31	6.96
47.60	-6.55	-0.60	334.35	18.33	-32.72	6.58
53.55	-12.05	-1.25	447.27	19.57	-25.45	-4.32
59.50	-21.37	-1.77	558.03	21.17	-65.74	-6.62
65.45	-31.89	-1.73	700.21	23.26	-80.84	2.27
71.40	-42.43	-1.95	843.81	24.67	-48.19	6.71
77.35	-56.99	-2.99	993.79	25.39	-23.48	0.45
84.15	-81.29	-4.07	1167.70	25.92	-48.27	-6.06

Since the capsule tends to pitch bow down when separated and move backwards, it will quite likely interfere with the remaining vehicle. Not a satisfactory situation.

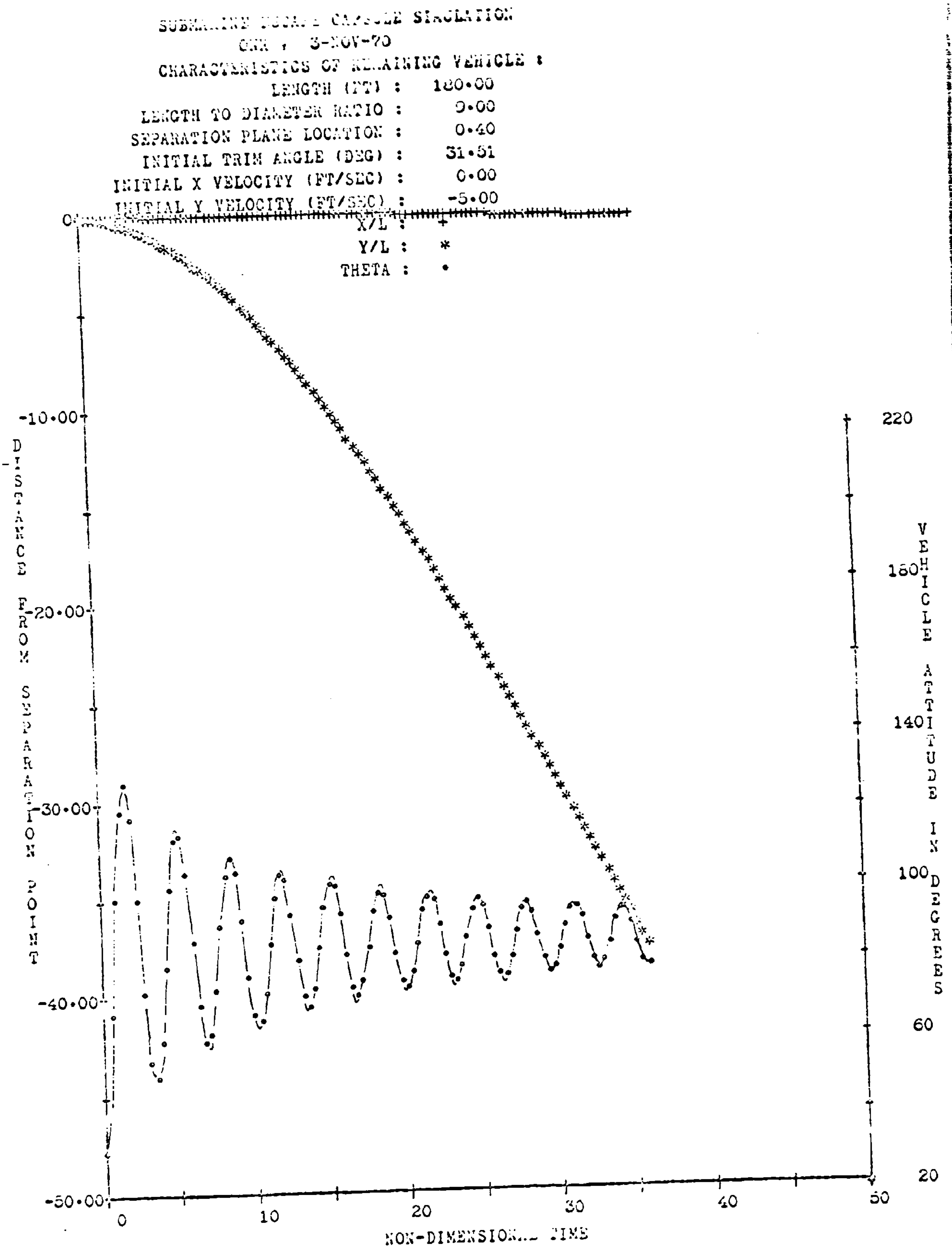
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --)

**HULL --calculate the trajectory of the remaining hull

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 85

**GRAPH

CHECK PLOT SWITCH


3-NOV-70
LENGTH (FT) : 180.00
LENGTH TO DIAMETER RATIO : 9.00
SEPARATION PLANE LOCATION : 0.40
INITIAL TRIM ANGLE (DEG) : 31.51
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : -5.00
X/L : +
Y/L : *
THETA : •
DISTANCE FROM SEPARATION POINT
-10.00
-20.00
-30.00
-40.00
-50.00
VEHICLE ATTITUDE IN DEGREES
220
180
140
100
60
20
0
10
20
30
40
50
NON-DIMENSIONAL TIME

RESET FLG. SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --)

**PRINT

HOW MANY POINTS DO YOU WISH TO PRINT (2 TO 14) -- 14

NON-DIMENSIONAL RESULTS

TIME	HORIZONTAL		VERTICAL		ATTITUDE	ANGULAR
	DISTANCE	VELOCITY	DISTANCE	VELOCITY	DEGREES	VELOCITY
0.00	0.00	0.00	0.00	-0.07	31.51	0.00
2.52	0.00	0.00	-0.39	-0.25	97.97	-69.71
5.03	-0.00	-0.00	-1.26	-0.44	113.38	22.07
7.55	-0.00	-0.00	-2.59	-0.62	73.60	42.22
10.06	-0.01	-0.00	-4.37	-0.79	67.28	-18.03
12.58	-0.03	-0.01	-6.54	-0.94	93.46	-32.13
15.09	-0.05	-0.01	-9.07	-1.07	101.03	8.57
17.61	-0.08	-0.02	-11.91	-1.18	84.06	27.49
20.12	-0.12	-0.02	-14.99	-1.27	73.18	2.44
22.64	-0.17	-0.02	-18.28	-1.34	82.24	-21.50
25.15	-0.22	-0.02	-21.73	-1.40	94.54	-11.86
27.67	-0.28	-0.03	-25.31	-1.44	93.26	11.89
30.18	-0.35	-0.03	-28.98	-1.48	82.49	16.72
32.70	-0.41	-0.03	-32.73	-1.50	77.51	-1.07
35.93	-0.51	-0.03	-37.63	-1.53	78.22	1.92

DIMENSIONALIZED RESULTS

TIME	HORIZONTAL		VERTICAL		ATTITUDE	ANGULAR
	DISTANCE	VELOCITY	DISTANCE	VELOCITY	DEGREES	VELOCITY
SECONDS	FEET	FT/SEC	FEET	FT/SEC		DEG/SEC
0.00	0.00	0.00	0.00	-5.00	31.51	0.00
5.95	0.04	0.05	-70.61	-18.97	97.97	-29.47
11.90	-0.30	-0.10	-226.43	-33.44	113.38	9.33
17.85	-0.80	-0.33	-466.43	-46.86	73.60	17.85
23.80	-2.26	-0.26	-785.91	-60.14	67.28	-7.62
29.75	-5.36	-0.42	-1177.45	-71.53	93.46	-13.58
35.70	-9.64	-0.87	-1633.18	-81.45	101.03	3.62
41.65	-15.00	-1.25	-2143.48	-89.69	84.06	11.62
47.60	-21.82	-1.32	-2698.94	-95.63	73.18	1.03
53.55	-30.39	-1.35	-3290.81	-102.13	82.24	-9.09
59.50	-40.21	-1.63	-3911.82	-106.46	94.54	-5.01
65.45	-50.37	-1.95	-4555.65	-109.80	93.26	5.03
71.40	-62.29	-2.15	-5217.06	-112.38	82.49	7.07
77.35	-[illegible].63	-2.12	-5831.87	-114.35	77.51	-0.45
83.00	-91.27	-2.26	-6774.05	-116.18	78.22	0.81

The remaining vehicle trims up suddenly and moves forward, making interference more probable.

**INPUT SEAS

INPUT SURFACE SEA STATE -- 6 ← Capsule surfaces in a state 6 sea

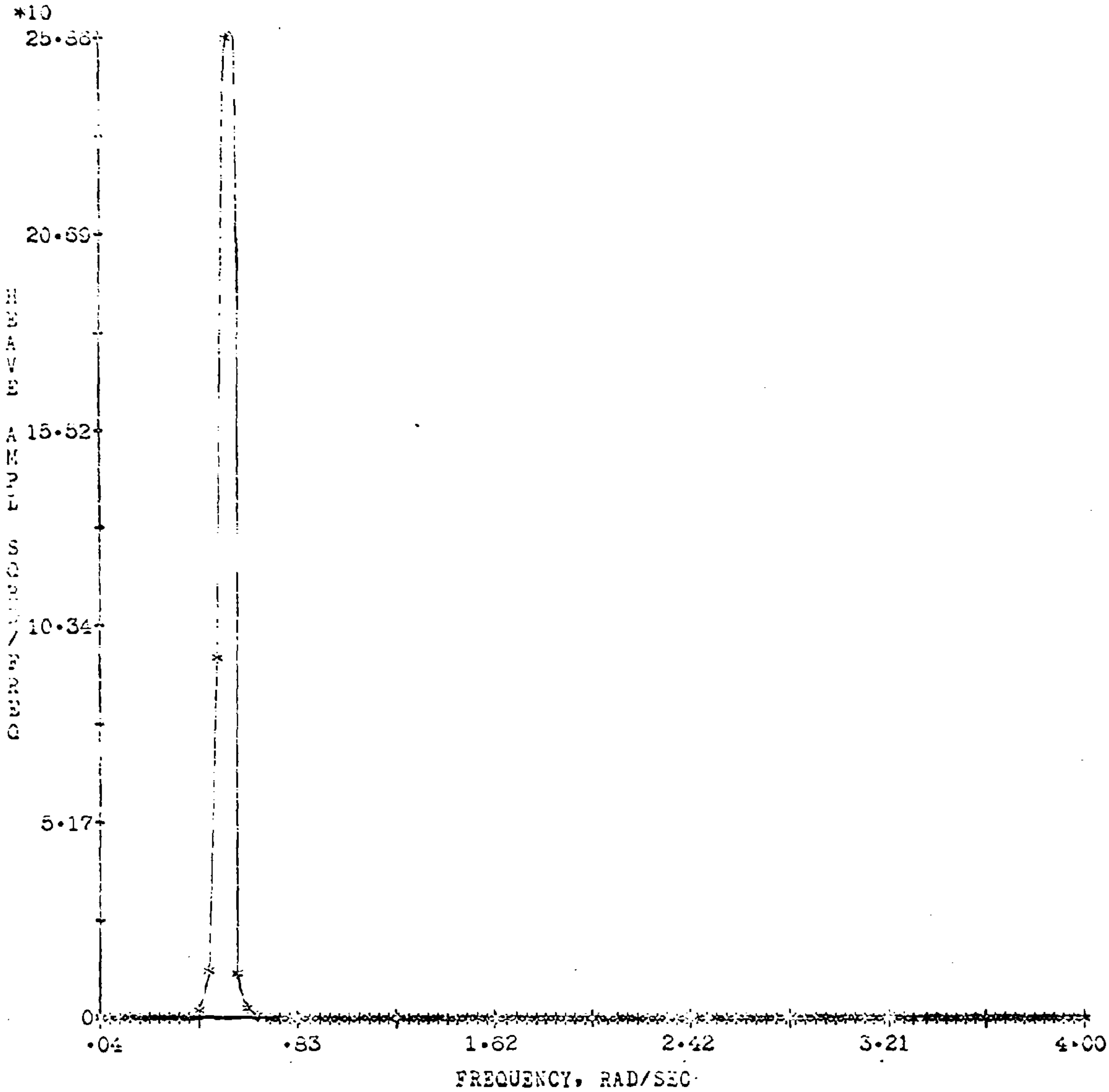
**MOTIONS --Calculates capsule response to a state 6 sea.

THE CAPSULE IS FLOATING BOW DOWN.

SURFACE SEA STATE IS 6

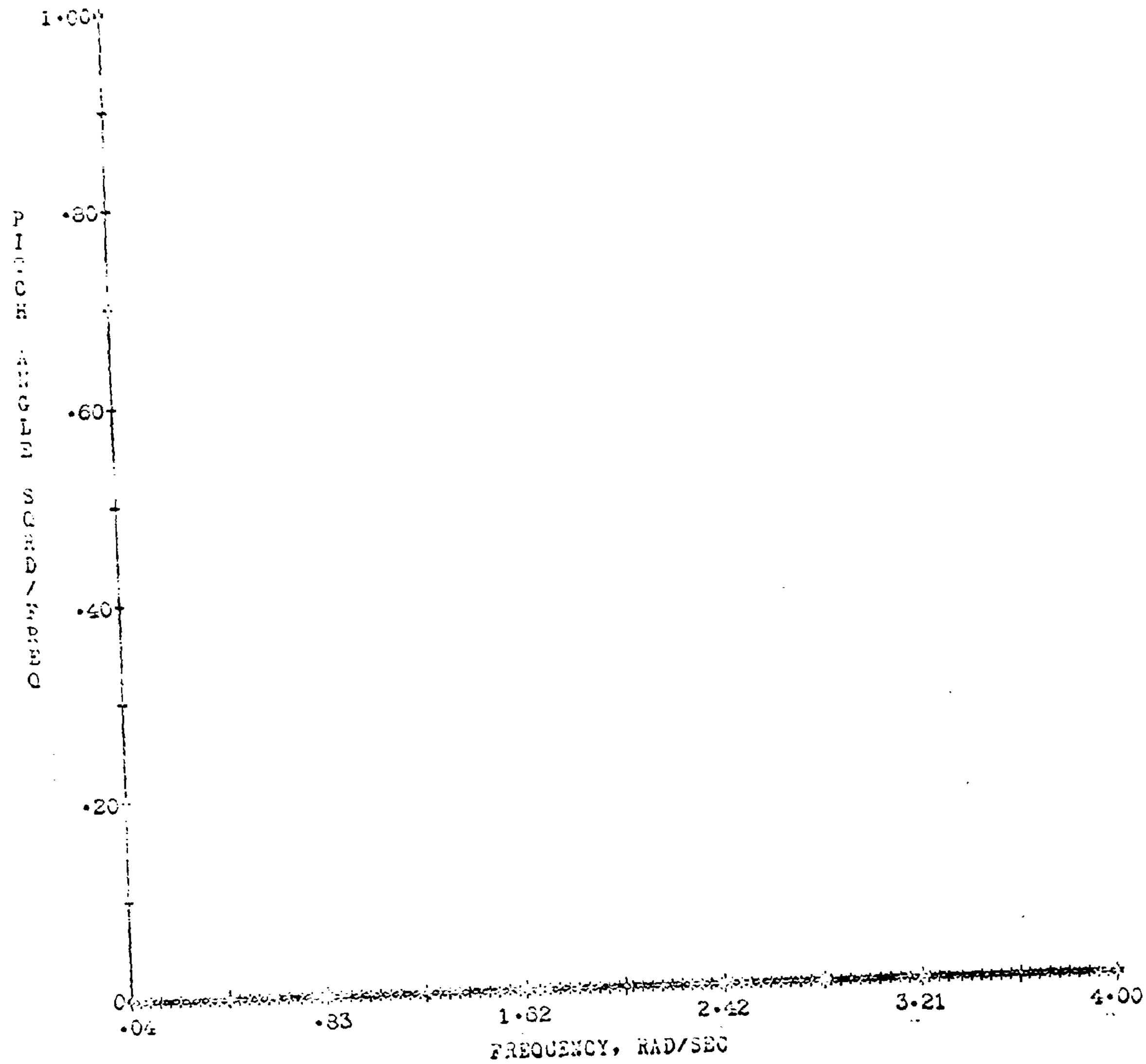
SIGNIFICANT WAVE HEIGHT = 12.542 FT
SIGNIFICANT HEAVE AMPL. = 15.634 FT
AVERAGE PITCH AMPL. = 3.871 DEGREES

CHECK PLOT SWITCH



*10
25.86
20.69
15.52
10.34
5.17
0
HEAVE AMPL SQRD/FREQ
.04
.83
1.62
2.42
3.21
4.00
FREQUENCY, RAD/SEC

CHECK PLOT SWITCH



```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN -- )

**END

EXECUTION TIME:         2 MIN. 51.52 SEC.
TOTAL ELAPSED TIME:     67 MIN. 4.82 SEC.
```

Complete simulation took 1 hour 7 minutes.

B. Simple Parametric Analysis

The second example of the use of the MODSEC system will demonstrate how moving the center of gravity and changing the buoyancy affects the trajectory of the ascending capsule.

As before we select a basic geometrical configuration. In this case we will consider a submarine 112 feet long with L/D of 5. We will locate the separation plane exactly in the middle of the submarine by letting $\lambda_o = .5$

We will use a very simple weight distribution so we can easily move the center of gravity. We assume the capsule shell is of zero thickness and the entire capsule can be considered as two lumped weights of 200 tons on the centerline, one located 0.2L from the bow, the other located at 0.4L from the bow.

Since we are only going to be concerned with the capsule motion we neglect any weights in the remaining vehicle.

We are now ready to run the MODSEC system.

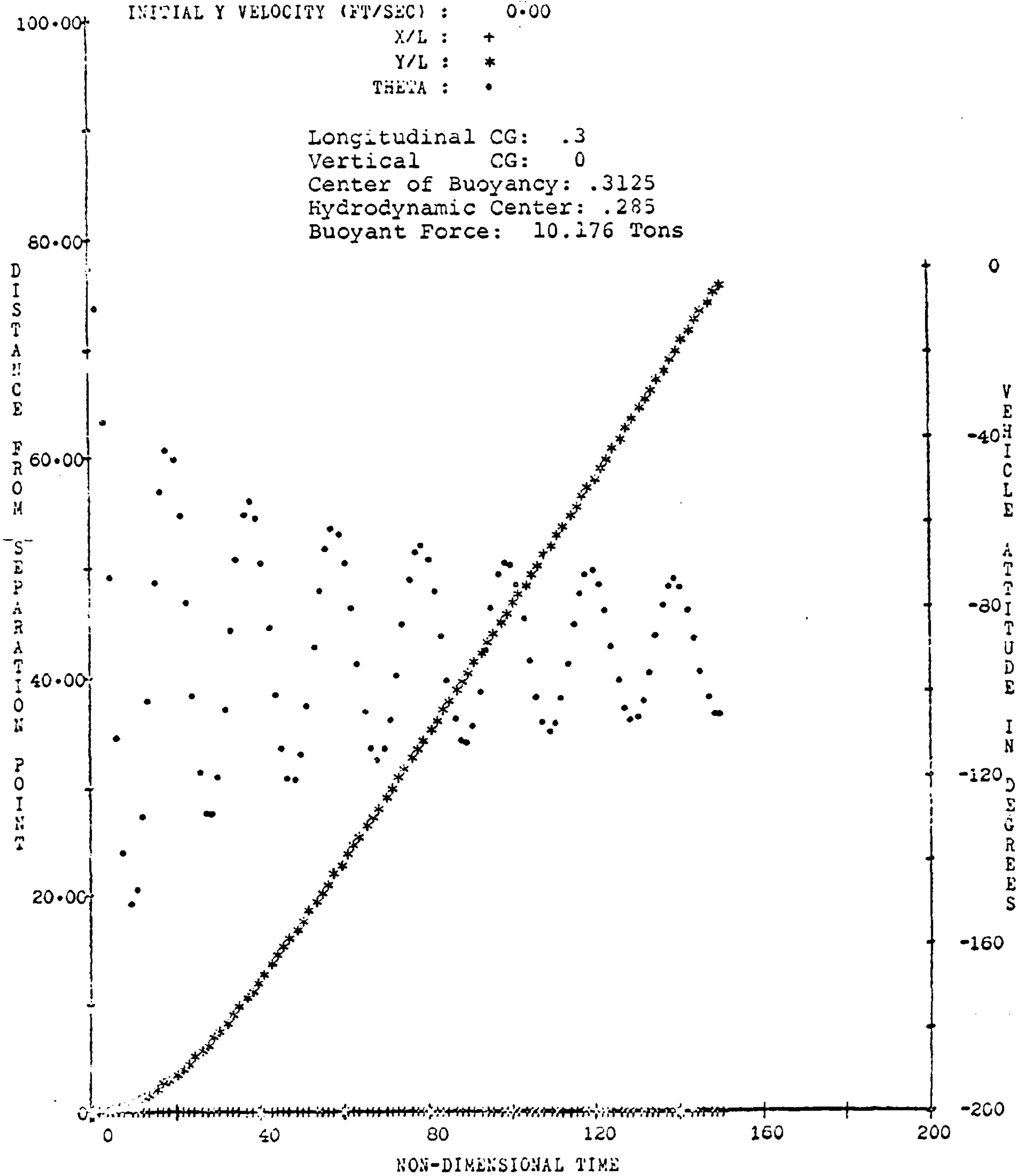
```
[illegible]
[illegible]

IF YOU REQUIRE INSTRUCTIONS, TYPE "HELP".
**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0. -- no shell weights

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) -- 0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- YES          -- mistake, did not want to use old weights

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAO.L (LBS/FT**3) -- ,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG'L CG, VERT CG
IN GROUPS OF THREE
200.,.2,.0
200.,.4,0

STORE WEIGHTS? (YES OR NO) -- NO

**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 5,112,.5

**WEIGHTS

**SEPARATE

YOU MUST INPUT AN INITIAL TRIM ANGLE --cannot separate without initial
**INPUT TRIM                           trim angle

SPECIFY INITIAL TRIM ANGLE (DEGREES) -- 0

**SEPARATE

WARNING--NO CASULTY SPECIFIED. DO YOU WANT TO CONTINUE? (YES OR NO) -- YES

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 200

**GRAPH
```

CHECK PLOT SWITCH

SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :

LENGTH (FT) :	56.00
LENGTH TO DIAMETER RATIO :	5.00
SEPARATION PLANE LOCATION :	0.50
INITIAL TRIM ANGLE (DEG) :	0.00
INITIAL X VELOCITY (FT/SEC) :	0.00
INITIAL Y VELOCITY (FT/SEC) :	0.00
X/L :	+
Y/L :	*
THETA :	•

Longitudinal CG: .3
Vertical CG: 0
Center of Buoyancy: .3125
Hydrodynamic Center: .285
Buoyant Force: 10.176 Tons



RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN -- ⮐

**INPUT GEOMETRY

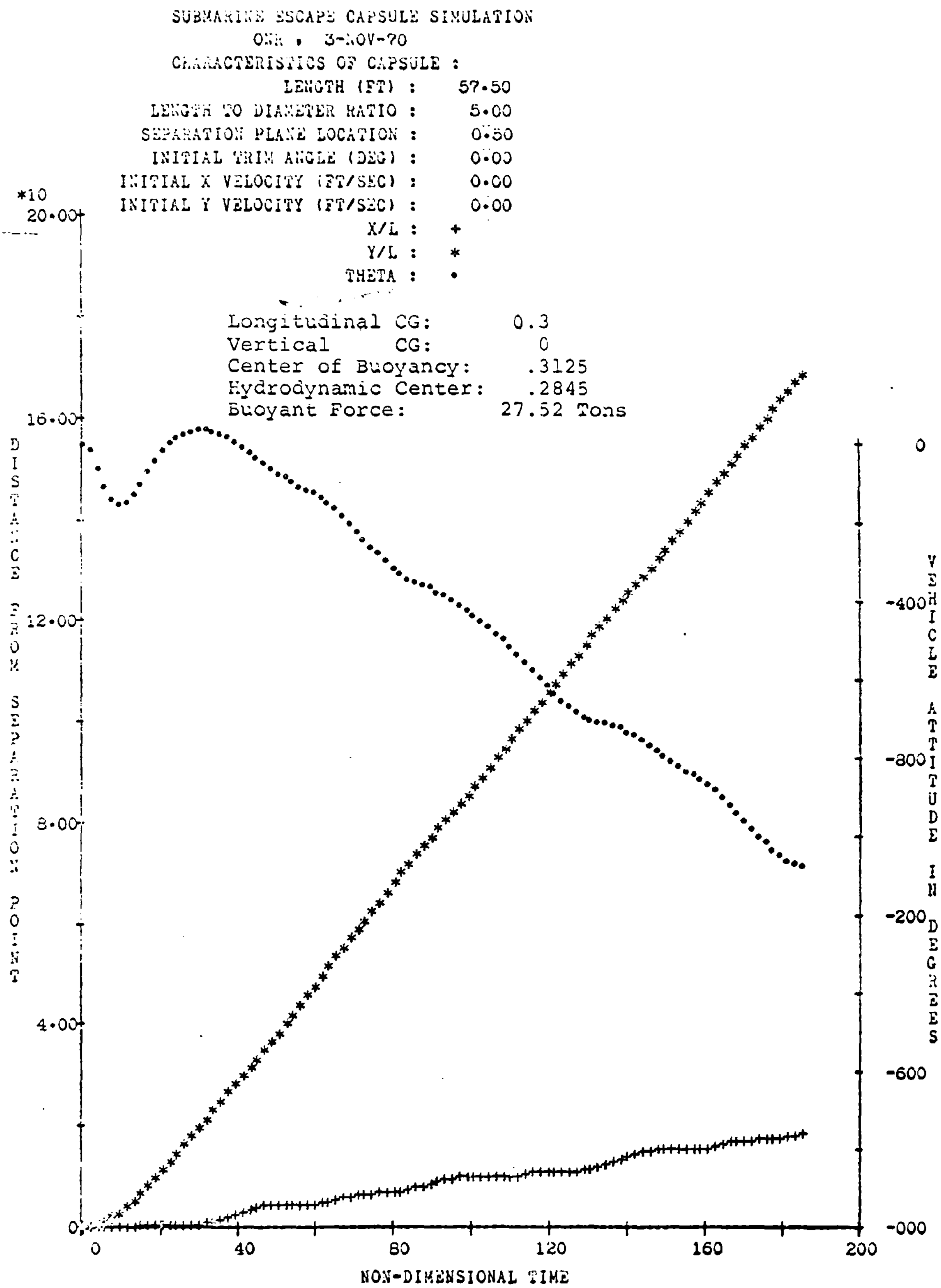
SPECIFY L/D, LENGTH, SEPARATION PLANE -- 5,115,.5

**ASCEND

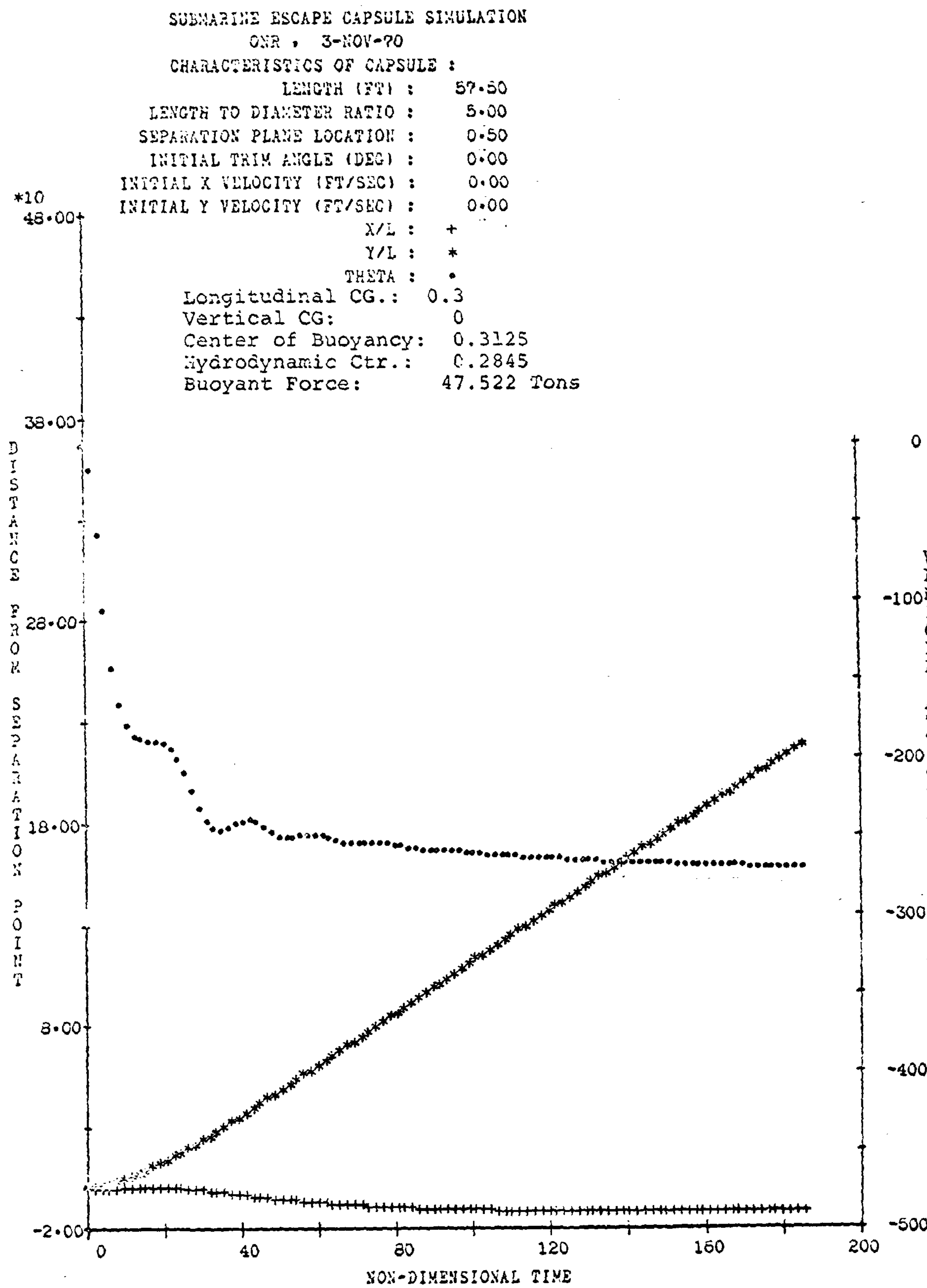
INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH

CHECK PLOT SWITCH

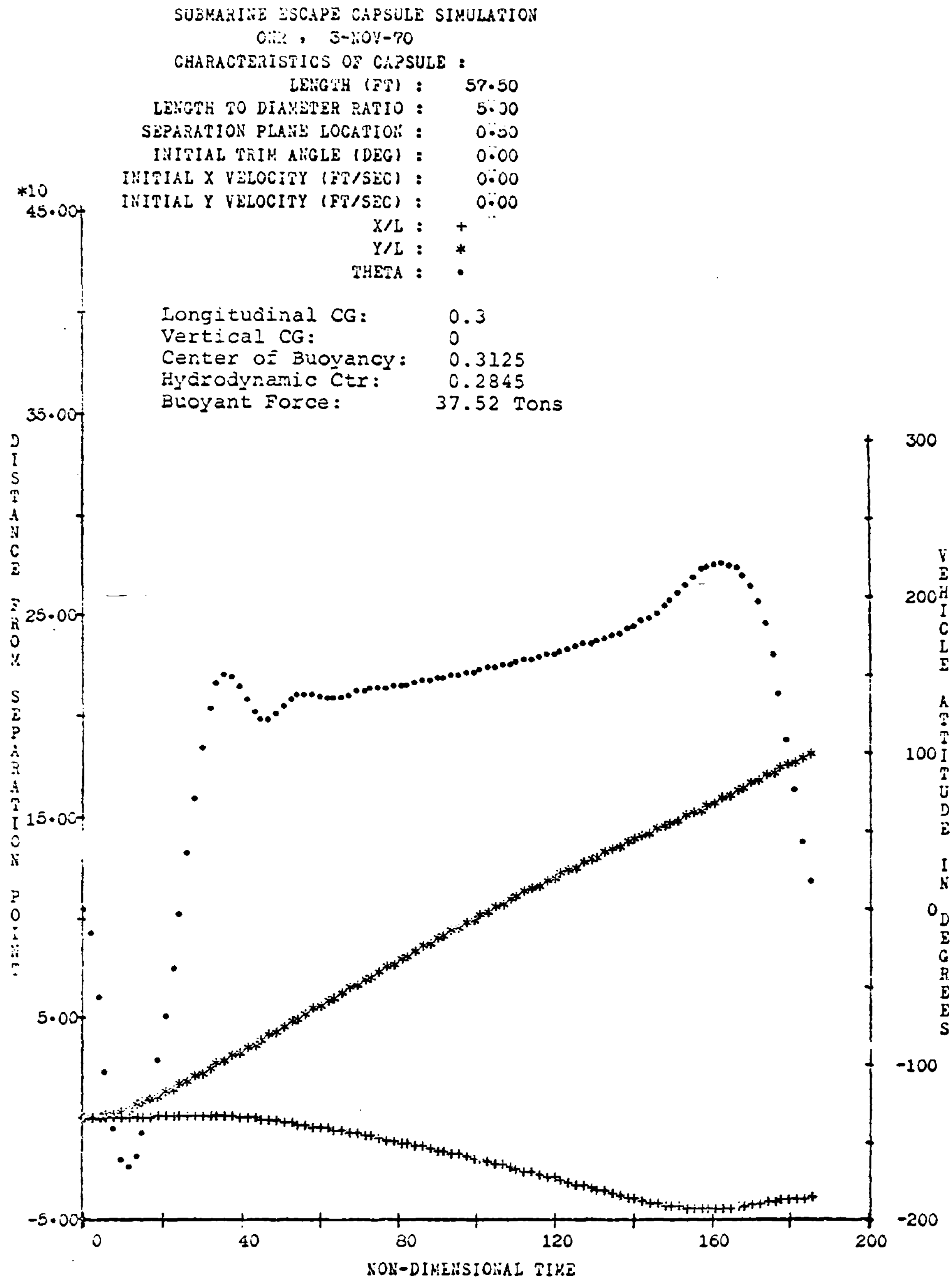

SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG: 0.3
Vertical CG: 0
Center of Buoyancy: .3125
Hydrodynamic Center: .2845
Buoyant Force: 27.52 Tons
*10
20.00
16.00
12.00
8.00
4.00
0
DISTANCE FROM SEPARATION POINT
0
-400
-800
-200
-600
-000
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN -- ↲

**INPUT WEIGHT

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG"L CG, VERT CG
IN GROUPS OF THREE
190,.2,0
190,.4,0

STORE WEIGHTS? (YES OR NO) -- NO

**WEIGHTS

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH


CHECK PLOT SWITCH
```



SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG.: 0.3
Vertical CG: 0
Center of Buoyancy: 0.3125
Hydrodynamic Ctr.: 0.2845
Buoyant Force: 47.522 Tons
*10
48.00
38.00
28.00
18.00
8.00
-2.00
DISTANCE FROM SEPARATION POINT
0
-100
-200
-300
-400
-500
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**INPUT WEIGHT

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG'L CG, VERT CG
IN GROUPS OF THREE
195.,2,0
195.,4,0

STORE WEIGHTS? (YES OR NO) -- NO

**WEIGHTS

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH


CHECK PLOT SWITCH
```


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG: 0.3
Vertical CG: 0
Center of Buoyancy: 0.3125
Hydrodynamic Ctr: 0.2845
Buoyant Force: 37.52 Tons
*10
45.00
35.00
25.00
15.00
5.00
-5.00
DISTANCE FROM SEPARATION POINT
300
200
100
0
-100
-200
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

To investigate the capsule behavior for a longer time we need only to increase the simulation time

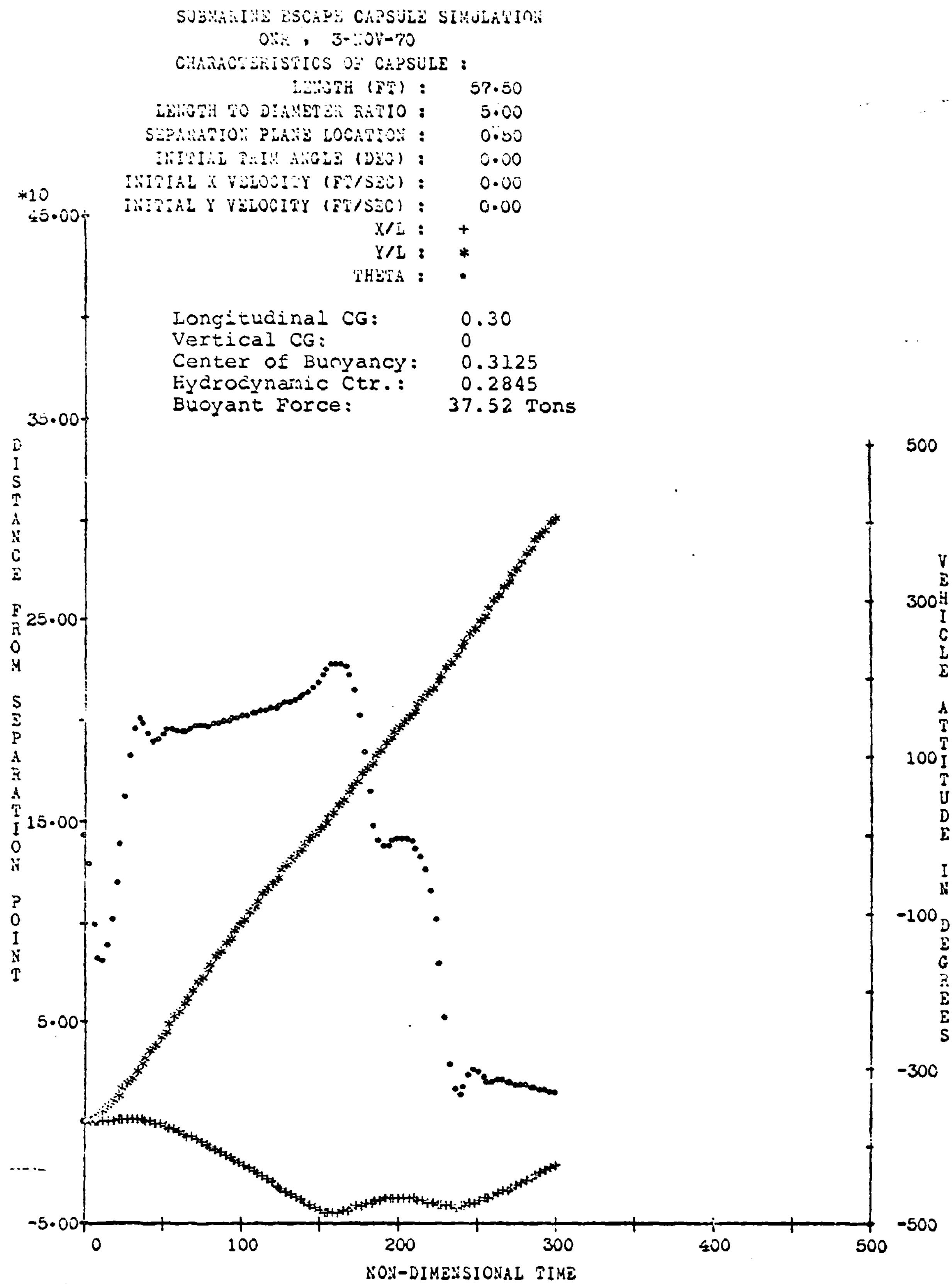
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN -- ⟩

**ASCEND

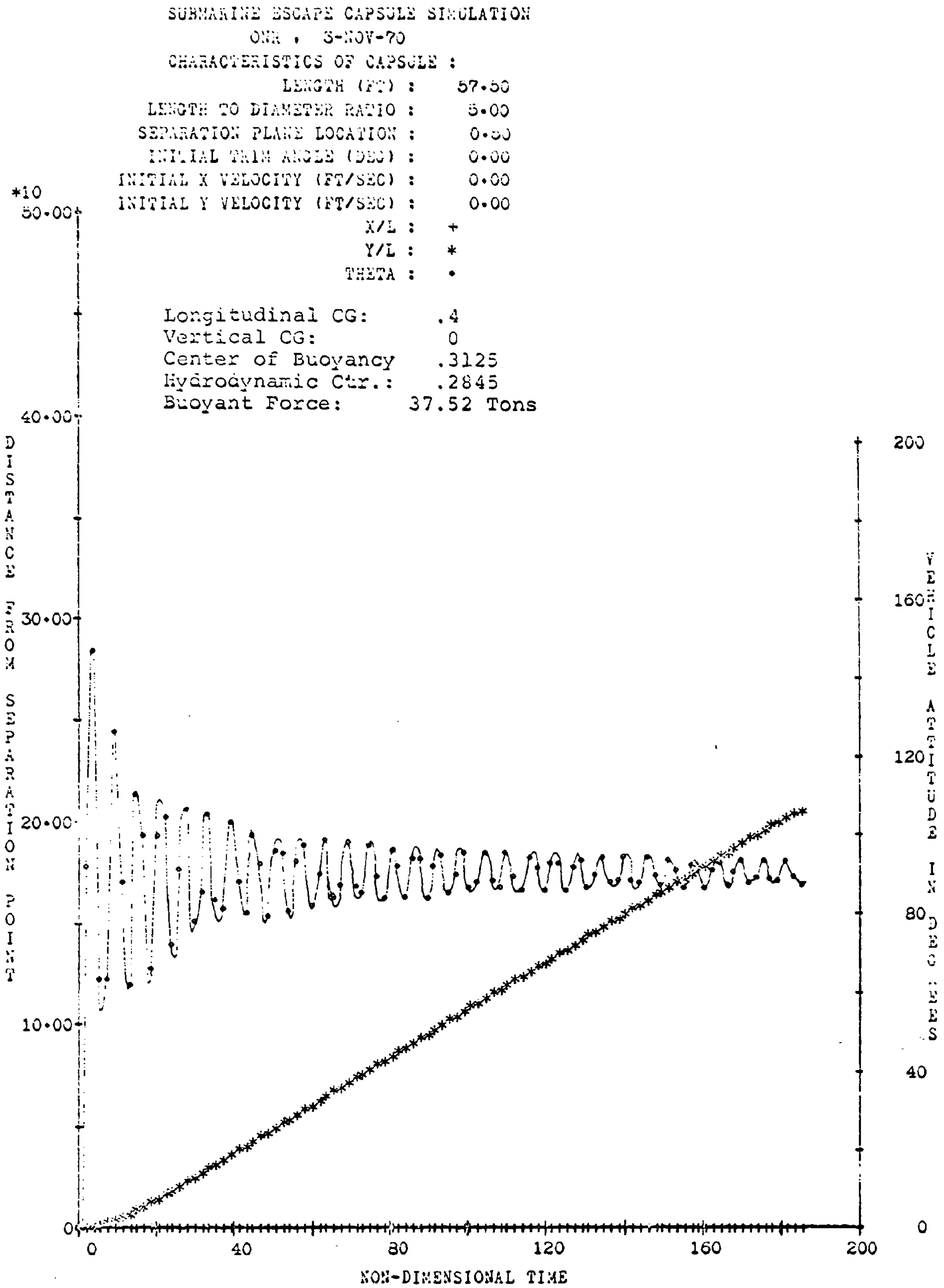
INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 400

**GRAPH

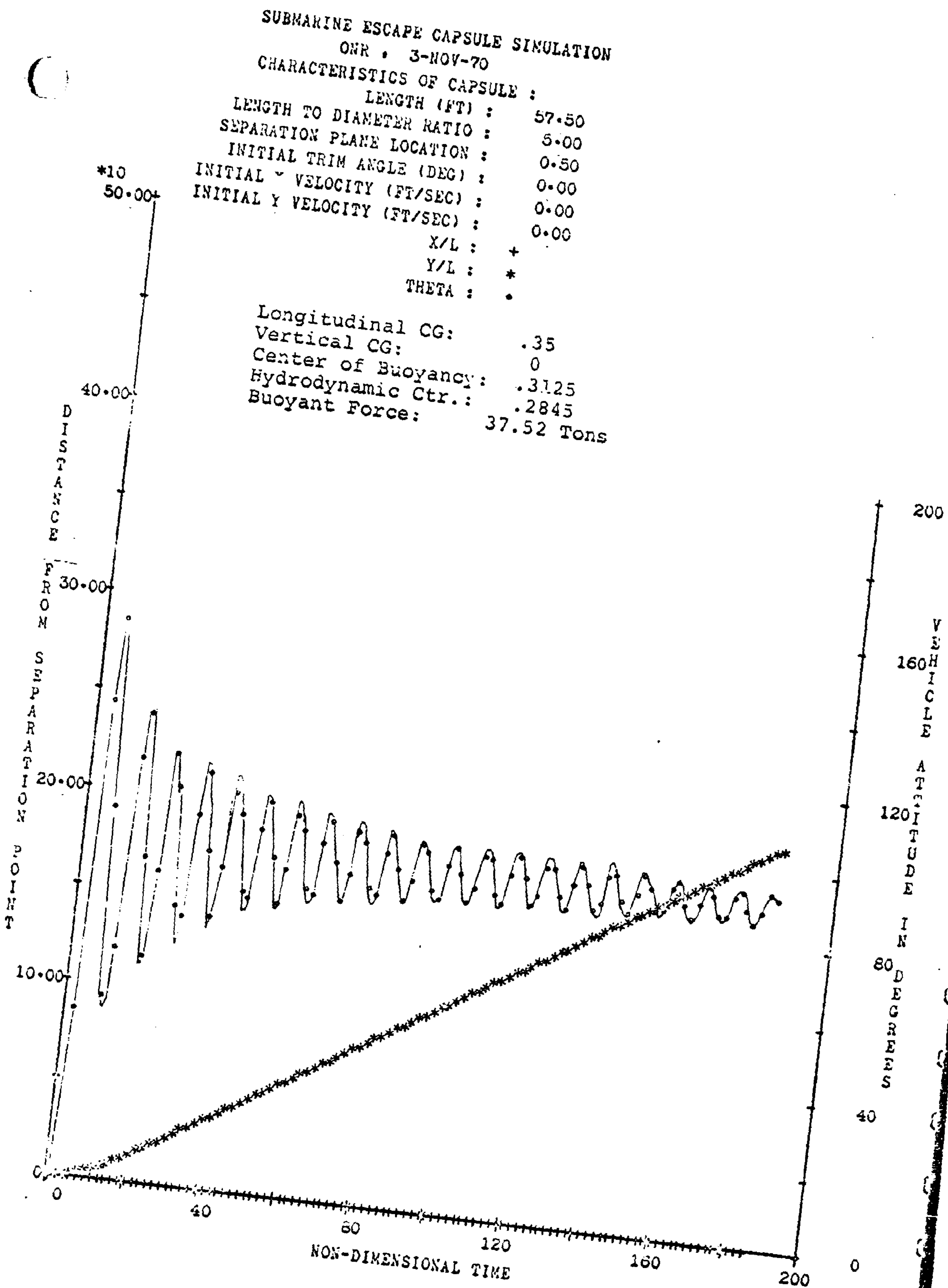
CHECK PLOT SWITCH


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG: 0.30
Vertical CG: 0
Center of Buoyancy: 0.3125
Hydrodynamic Ctr.: 0.2845
Buoyant Force: 37.52 Tons
*10
45.00
35.00
25.00
15.00
5.00
-5.00
DISTANCE FROM SEPARATION POINT
500
300
100
-100
-300
-500
VEHICLE ATTITUDE IN DEGREES
0
100
200
300
400
500
NON-DIMENSIONAL TIME

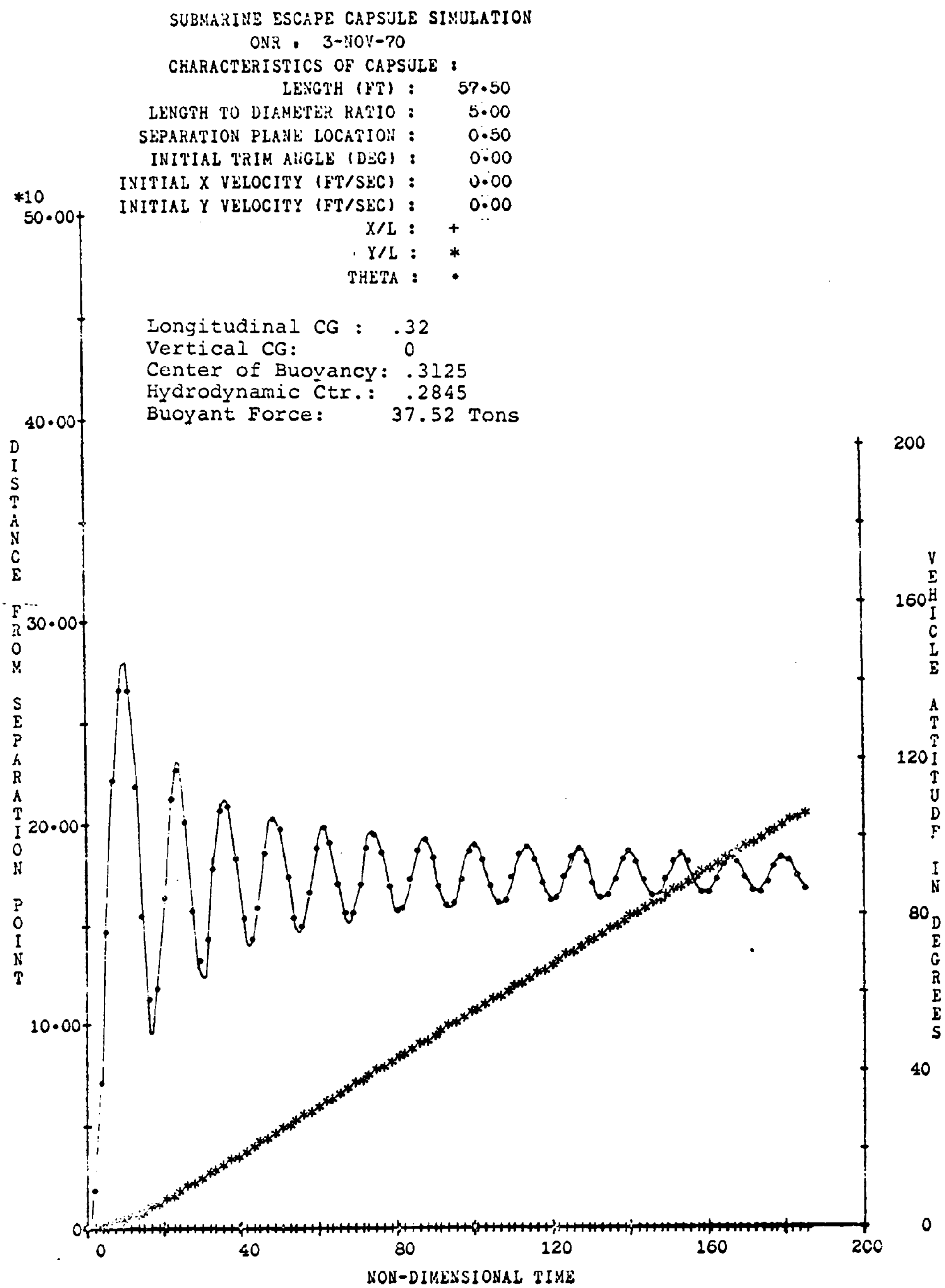
```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG"L CG, VERT CG
IN GROUPS OF THREE
195,-3,0
195,-49,0

STORE WEIGHTS? (YES OR NO) -- NO

**WEIGHTS

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH


CHECK PLOT SWITCH
```


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG: .4
Vertical CG: 0
Center of Buoyancy .3125
Hydrodynamic Ctr.: .2845
Buoyant Force: 37.52 Tons
*10
50.00
40.00
30.00
20.00
10.00
0
DISTANCE FROM SEPARATION POINT
200
160
120
80
40
0
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG'L CG, VERT CG
IN GROUPS OF THREE
195,.25,0
195,.45,0

STORE WEIGHTS? (YES OR NO) -- NO

**WEIGHTS

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH


CHECK PLOT SWITCH
```


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG: .35
Vertical CG: 0
Center of Buoyancy: .3125
Hydrodynamic Ctr.: .2845
Buoyant Force: 37.52 Tons
*10
50.00
40.00
30.00
20.00
10.00
0
DISTANCE FROM SEPARATION POINT
0
40
80
120
160
200
NON-DIMENSIONAL TIME
200
160
120
80
40
0
VEHICLE ATTITUDE IN DEGREES

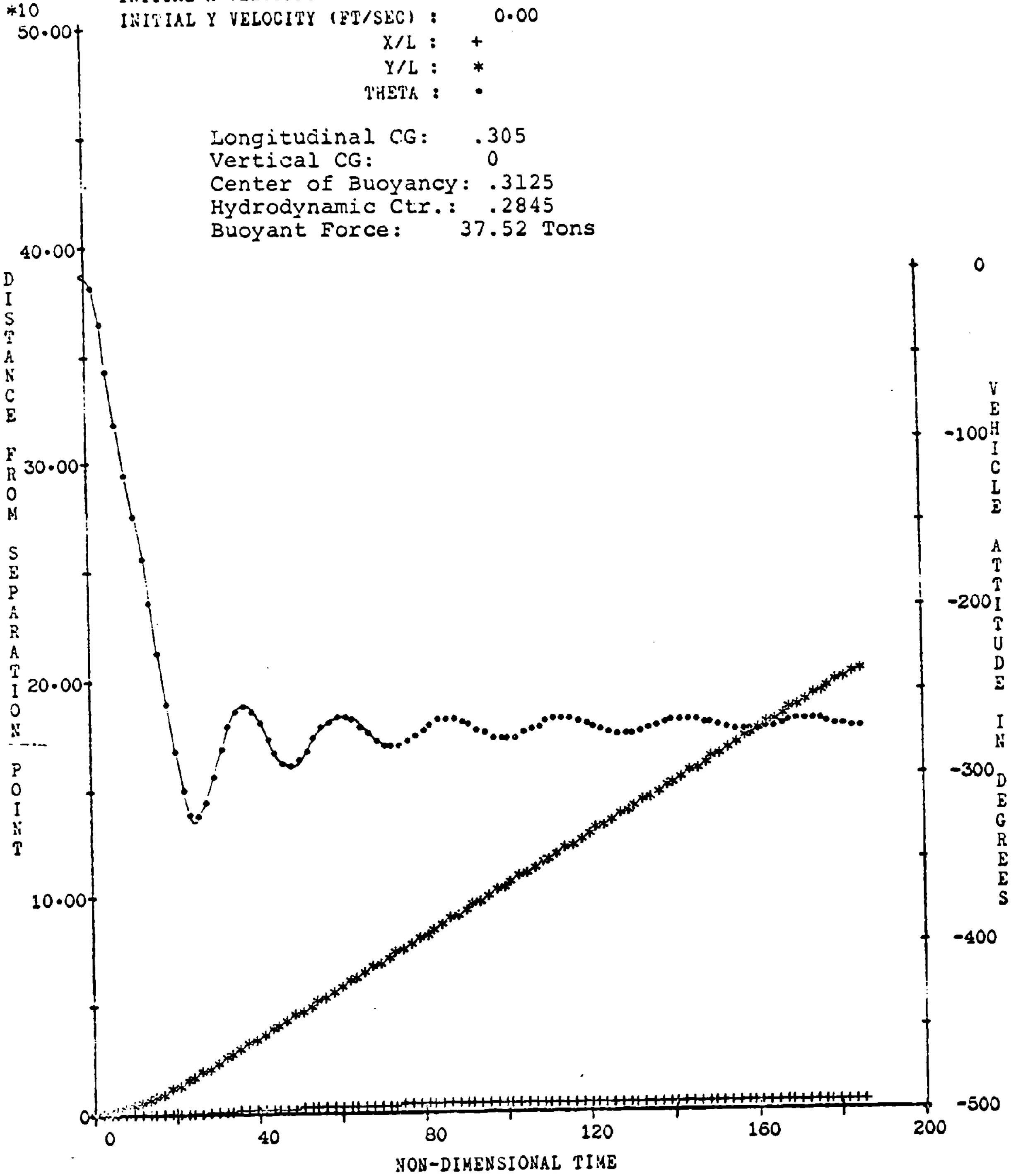
```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN -- }

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG"L CG, VERT CG
IN GROUPS OF THREE
195,.22,0
195,.42,0

STORE WEIGHTS? (YES OR NO) -- NO

**WEIGHTS

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH


CHECK PLOT SWITCH
```


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR . 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
Longitudinal CG : .32
Vertical CG: 0
Center of Buoyancy: .3125
Hydrodynamic Ctr.: .2845
Buoyant Force: 37.52 Tons
*10
50.00
40.00
30.00
20.00
10.00
0
DISTANCE FROM SEPARATION POINT
200
160
120
80
40
0
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN -- )

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0.,0.

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0.,0.

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 2

INPUT WEIGHTS IN LONG TONS, LONG'L CG, VERT CG
IN GROUPS OF THREE
195,.19,0
195,.42,0

STORE WEIGHTS? (YES OR NO) -- NO

**WEIGHTS

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 250

**GRAPH


CHECK PLOT SWITCH
```

SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :

LENGTH (FT) :	57.50
LENGTH TO DIAMETER RATIO :	5.00
SEPARATION PLANE LOCATION :	0.50
INITIAL TRIM ANGLE (DEG) :	0.00
INITIAL X VELOCITY (FT/SEC) :	0.00
INITIAL Y VELOCITY (FT/SEC) :	0.00
X/L :	+
Y/L :	*
THETA :	•



```
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**END

EXECUTION TIME:          2 MIN. 55.13 SEC.
TOTAL ELAPSED TIME:     83 MIN. 13.73 SEC.
```

C. Simple Check On Capsule and Hull Behavior

If the submarine is exactly symmetrical, the capsule should behave just like the remaining vehicle. That is, if the capsule rises, moves forward and pitches up, the remaining vehicle should rise move backward, and pitch down. In other words, there should be a vertical plane of symmetry for the motion.

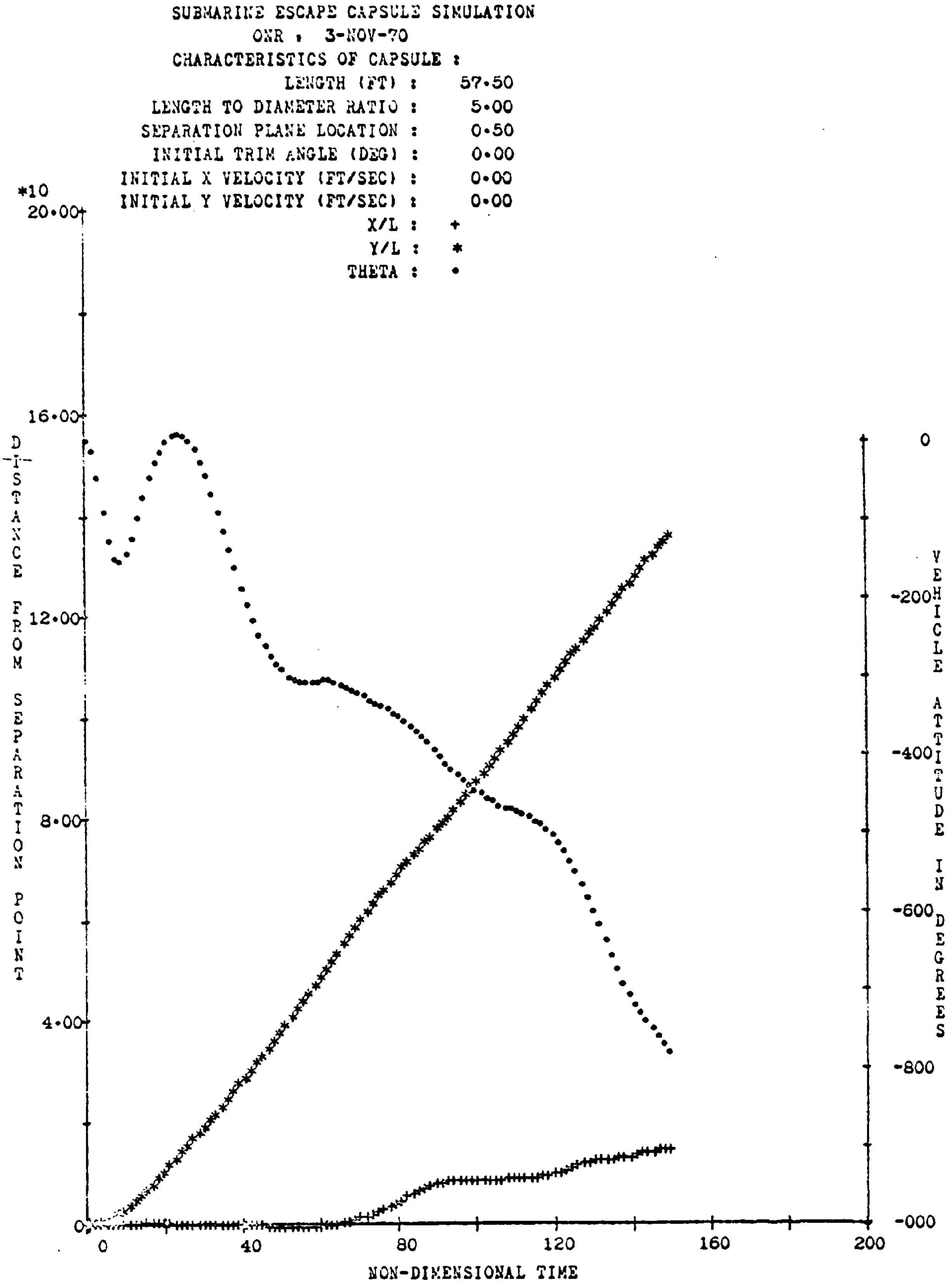
In order to check this, the same geometry and weight distribution used in The Simple Parametric Analysis will be used. Since we are now concerned with the remaining vehicle motions as well as the capsule trajectory, two more 200-ton weights will be input on the centerline at 0.6L and 0.8L. The L/D is 5, the length is 115 feet and the separation plane is located at 0.5L.

The graphical output confirms the expected symmetrical behavior.

Note that the command abbreviations are used extensively in this example.

```
.RUN DSK:MODSEC.SAV
EXECUTION

IF YOU REQUIRE INSTRUCTIONS, TYPE "HELP".
**INPUT GEOMETRY

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 5,100,.5

**INPUT WEIGHTS

INPUT AVERAGE CAPSULE SHELL THICKNESS (IN), AVERAGE
HULL SHELL THICKNESS (IN) -- 0,0

INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND
OF HULL MATERIAL (LBS/FT**3) --  0,0

DO YOU WANT TO USE OLD WEIGHTS?
YES OR NO -- NO

HOW MANY WEIGHTS ARE TO BE INPUT? -- 4

INPUT WEIGHTS IN LONG TONS, LONG'L CG, VERT CG
IN GROUPS OF THREE
200,.2,0
200,.4,0
200,.6,0
200,.8,0

STORE WEIGHTS? (YES OR NO) -- NO

**TRIM

TRIM ANGLE CANNOT BE FOUND UNTIL WEIGHTS AND CENTERS ARE FOUND
**WEIGHTS

**TRIM

INTACT VEHICLE IS HEAVY OVERALL BY    451588. POUNDS
CG AND CB COINCIDENT, TRIM ANGLE INDETERMINATE

**I G

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 5,108,.5

**T

INTACT VEHICLE IS HEAVY OVERALL BY    103467. POUNDS
CG AND CB COINCIDENT, TRIM ANGLE INDETERMINATE

**I G

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 5,110,.5
```

```
**TRIM

INTACT VEHICLE IS HEAVY OVERALL BY      7912. POUNDS
CG AND CB COINCIDENT, TRIM ANGLE INDETERMINATE

**I G

SPECIFY L/D, LENGTH, SEPARATION PLANE -- 5,115,.5

**TRIM

INTACT VEHICLE IS LIGHT OVERALL BY    246599. POUNDS
CG AND CB COINCIDENT, TRIM ANGLE INDETERMINATE

**SEPARATE

YOU MUST INPUT AN INITIAL TRIM ANGLE
**INPUT TRIM

SPECIFY INITIAL TRIM ANGLE (DEGREES) -- 0

**SEPARATE

WARNING--NO CASULTY SPECIFIED. DO YOU WANT TO CONTINUE? (YES OR NO) -- YES

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 200

**GRAPH


CHECK PLOT SWITCH
```


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
*10
20.00
16.00
12.00
8.00
4.00
0
DISTANCE FROM SEPARATION POINT
0
-200
-400
-600
-800
-000
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

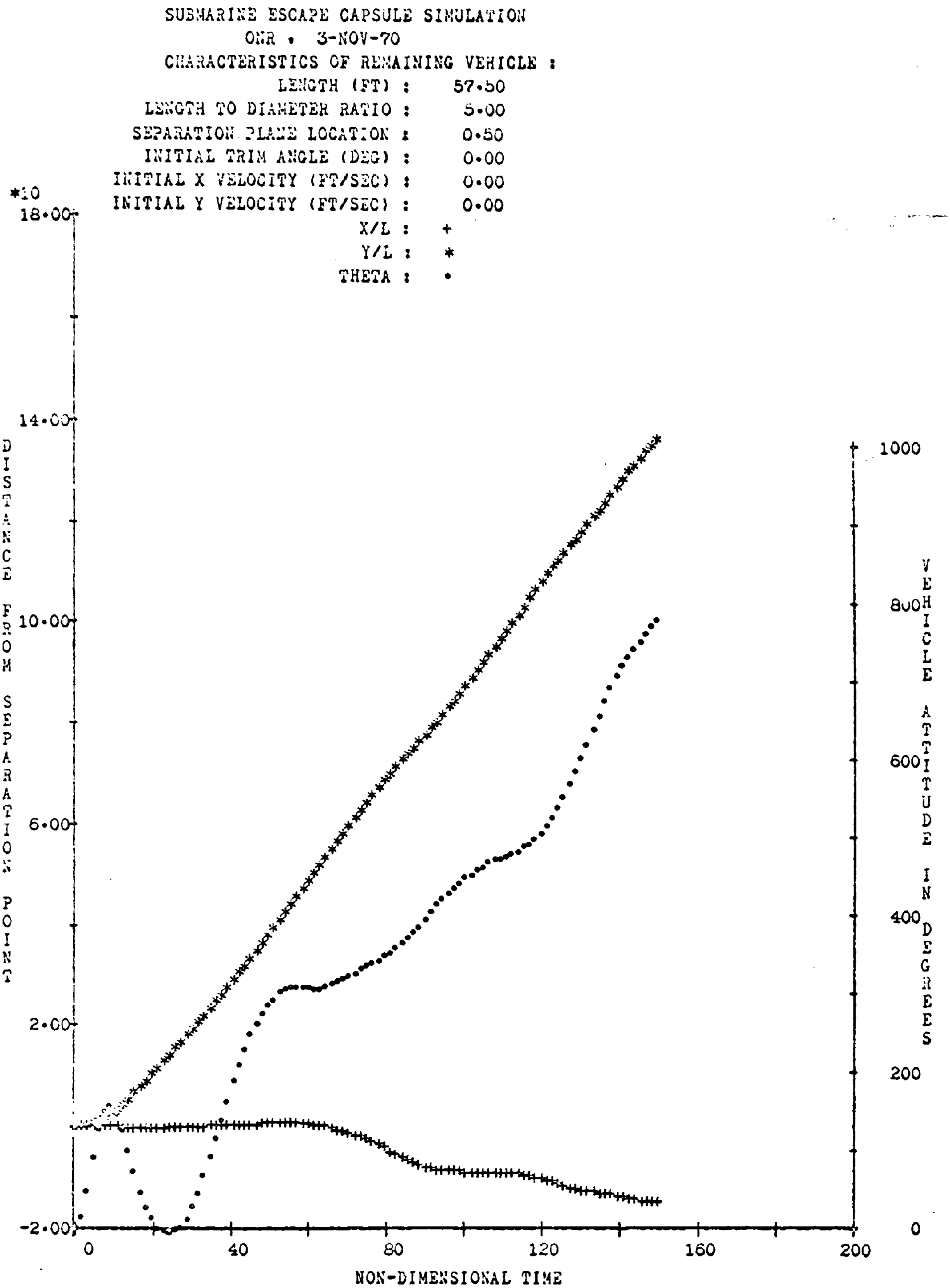
RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**HULL

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 200

**GRAPH

CHECK PLOT SWITCH


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR . 3-NOV-70
CHARACTERISTICS OF REMAINING VEHICLE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 0.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
*10
18.00
14.00
10.00
6.00
2.00
-2.00
DISTANCE FROM SEPARATION POINT
1000
800
600
400
200
0
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE
RETURN --

**INPUT SPEED

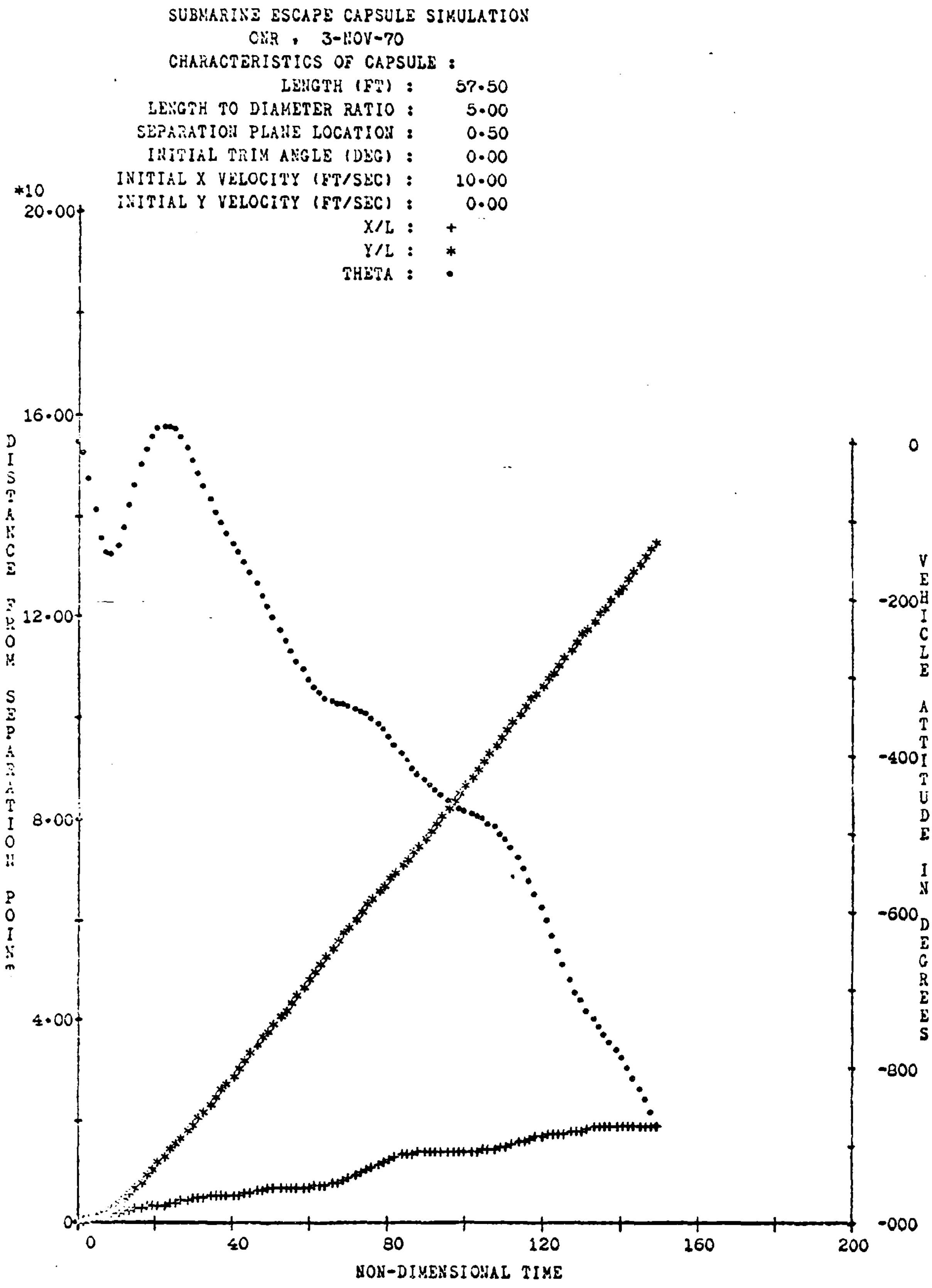
SPECIFY INITIAL VERTICAL AND HORIZONTAL SPEED (FT/SEC) -- 0,10

**ASCEND

INPUT MAXIMUM SIMULATION TIME IN SECONDS -- 200

**GRAPH

CHECK PLOT SWITCH


SUBMARINE ESCAPE CAPSULE SIMULATION
ONR , 3-NOV-70
CHARACTERISTICS OF CAPSULE :
LENGTH (FT) : 57.50
LENGTH TO DIAMETER RATIO : 5.00
SEPARATION PLANE LOCATION : 0.50
INITIAL TRIM ANGLE (DEG) : 0.00
INITIAL X VELOCITY (FT/SEC) : 10.00
INITIAL Y VELOCITY (FT/SEC) : 0.00
X/L : +
Y/L : *
THETA : •
*10
20.00
16.00
12.00
8.00
4.00
0
DISTANCE FROM SEPARATION POINT
0
-200
-400
-600
-800
-000
VEHICLE ATTITUDE IN DEGREES
0
40
80
120
160
200
NON-DIMENSIONAL TIME

RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**CHECK

PRESENT SIMULATION STATUS

GEOMETRICAL PARAMETERS HAVE BEEN CALCULATED

WEIGHTS AND CENTERS HAVE BEEN CALCULATED

INTACT VEHICLE ATTITUDE HAS BEEN CALCULATED

CAPSULE ACCENT HAS BEEN CALCULATED

LENGTH/DIAMETER RATIO	=	5.00
TOTAL VEHICLE LENGTH(FT)	=	115.00
SEPARATION PLANE LOCATION	=	0.50
CAPSULE SHELL THICKNESS(IN.)	=	0.00
HULL SHELL THICKNESS(IN.)	=	0.00
INITIAL HORIZONTAL VELOCITY	=	10.00
INITIAL VERTICAL VELOCITY	=	0.00
INITIAL DEPTH(FT)	=	0.00
INITIAL TRIM ANGLE(DEGREES)	=	0.00
LCG INTACT VEHICLE	=	0.500
VCG INTACT VEHICLE	=	0.000
LCG CAPSULE	=	0.300
VCG CAPSULE	=	0.000

DO YOU WISH TO INSPECT THE CURRENT WEIGHT DISTRIBUTION
TYPE Y OR N - Y

CURRENT WEIGHT DISTRIBUTION IS:

200.00	0.20	0.00
200.00	0.40	0.00
200.00	0.60	0.00
200.00	0.80	0.00

DO YOU WISH TO INSPECT THE CURRENT VOLUME DISTRIBUTION?
TYPE Y OR N - NO

**INPUT SEAS

INPUT SURFACE SEA STATE -- 5

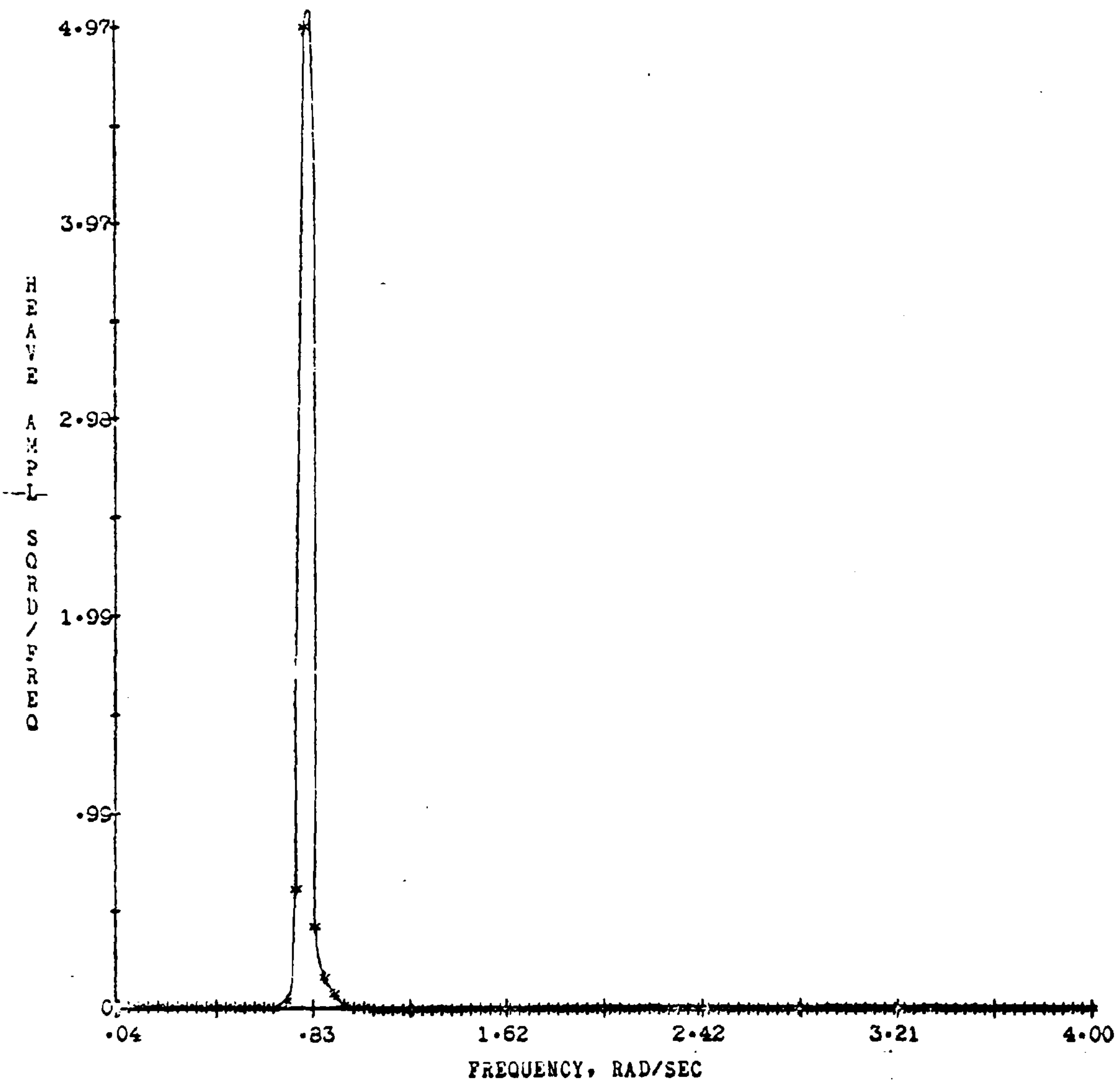
**MOTIONS

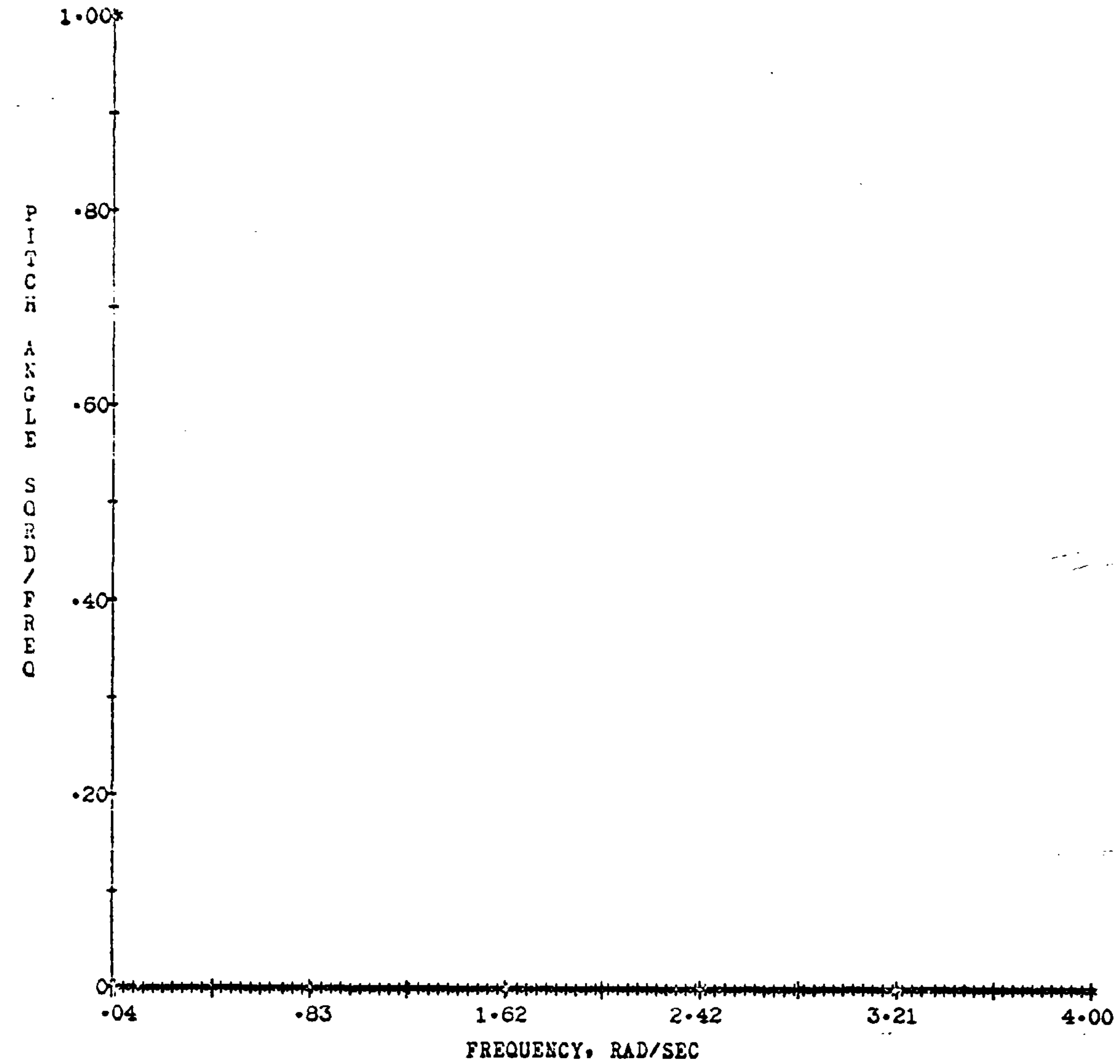
THE CAPSULE IS FLOATING BOW DOWN.

SURFACE SEA STATE IS 5

SIGNIFICANT WAVE HEIGHT = 3.856 FT
SIGNIFICANT HEAVE AMPL. = 2.015 FT
AVERAGE PITCH AMPL. = 1.069 DEGREES

CHECK PLOT SWITCH





RESET PLOT SWITCH TO TEXT MODE AND GIVE A CARRIAGE RETURN --

**END

EXECUTION TIME: 2 MIN. 51.05 SEC.
TOTAL ELAPSED TIME: 35 MIN. 7.30 SEC.

IX. COMPUTER PROGRAM DOCUMENTATION

IX-1. Main Program

The main program will be referred to hereafter as the monitor. It provides an interface for all of the subroutines, accepts commands, accepts data from the terminal, prints error messages, and maintains a status array to assist the user in performing a simulation in a logical fashion.

The monitor primarily uses the reserved COMMON areas as the medium for transferring input/output data between itself and a subroutine. This allows many of the subroutines to be called without using any subroutine arguments.

The monitor notifies the user that it is ready for a command by printing a double asterisk (**). Similarly the user is expected to respond with alpha or numeric data (within the context of the question) when a double dash (--) is encountered. Each monitor command is scanned for accuracy, and if a command is not recognized, a message so indicating is printed, and a double asterisk is printed indicating that the user is expected to input the proper command.

Since several subroutines require, as input, data that are computed by other subroutines, a sequential execution hierarchy must be followed by the user. An array has been set aside, ITEST (30), to contain the status of those subroutines which must be executed sequentially. Before a subroutine further in the logical sequence can be run, a test is made of the appropriate pre-requisite subroutines. If all conditions are met, execution continues; if not, an error message explaining the deficiency is printed at the terminal, and the user is returned to the double asterisk.

Error Messages

1) "WEIGHTS AND CENTERS CANNOT BE FOUND UNTIL GEOMETRY IS SPECIFIED" - The user has attempted to execute the subroutine which computes the weights and centers before specifying the geometry of the vehicle. Corrective action: specify vehicle geometry (i.e., INPUT GEOMETRY}).

2) "CENTERS CANNOT BE FOUND UNTIL THE WEIGHTS HAVE BEEN INPUT" - The user tried to calculate the longitudinal and vertical centers of gravity (i.e., WEIGHTS}) but he had not specified a weight distribution. Corrective action: input a weight distribution through keyboard or

specify a previously saved distribution through the INPUT WEIGHTS command and then give the command WEIGHTS}.

3) "YOU HAVE NOT INPUT A LIST OF FLOODABLE VOLUMES" - The user requested a damage simulation (DAMAGE}) but has not yet specified the floodable volume list. Corrective action: type INPUT VOLUME} and specify the floodable volumes.

4) "VEHICLE HAS ALREADY SEPARATED" - User requested the hydrostatic characteristics of the intact vehicle (TRIM}) subsequent to commanding SEPARATE}. Corrective action: run NEW CASE}.

5) "TRIM ANGLE CANNOT BE FOUND UNTIL WEIGHTS AND CENTERS ARE FOUND" - User attempted to find trim of intact vehicle (TRIM}) before specifying the weight array. Corrective action: use the INPUT WEIGHTS} command and try again.

6) "YOU MUST INPUT AN INITIAL TRIM ANGLE" - User gave the SEPARATE} command but failed to specify a trim angle. Corrective action: indicate trim angle by commanding INPUT TRIM} and then give the SEPARATE} command.

7) "WARNING -- NO CASUALTY SPECIFIED. DO YOU WANT TO CONTINUE? (YES OR NO)" - This is not an error message. It is printed to remind the user that he has not yet damaged the intact vehicle. He has the option of continuing (answer - YES}) or of specifying a casualty (answer - NO}, then give DAMAGE} command).

8) "CAPSULE HAS NOT BEEN SEPARATED FROM MAIN HULL" - The command ASCENT} or HULL} has been given before the SEPARATE} command. Corrective action: give the SEPARATE} command and try again.

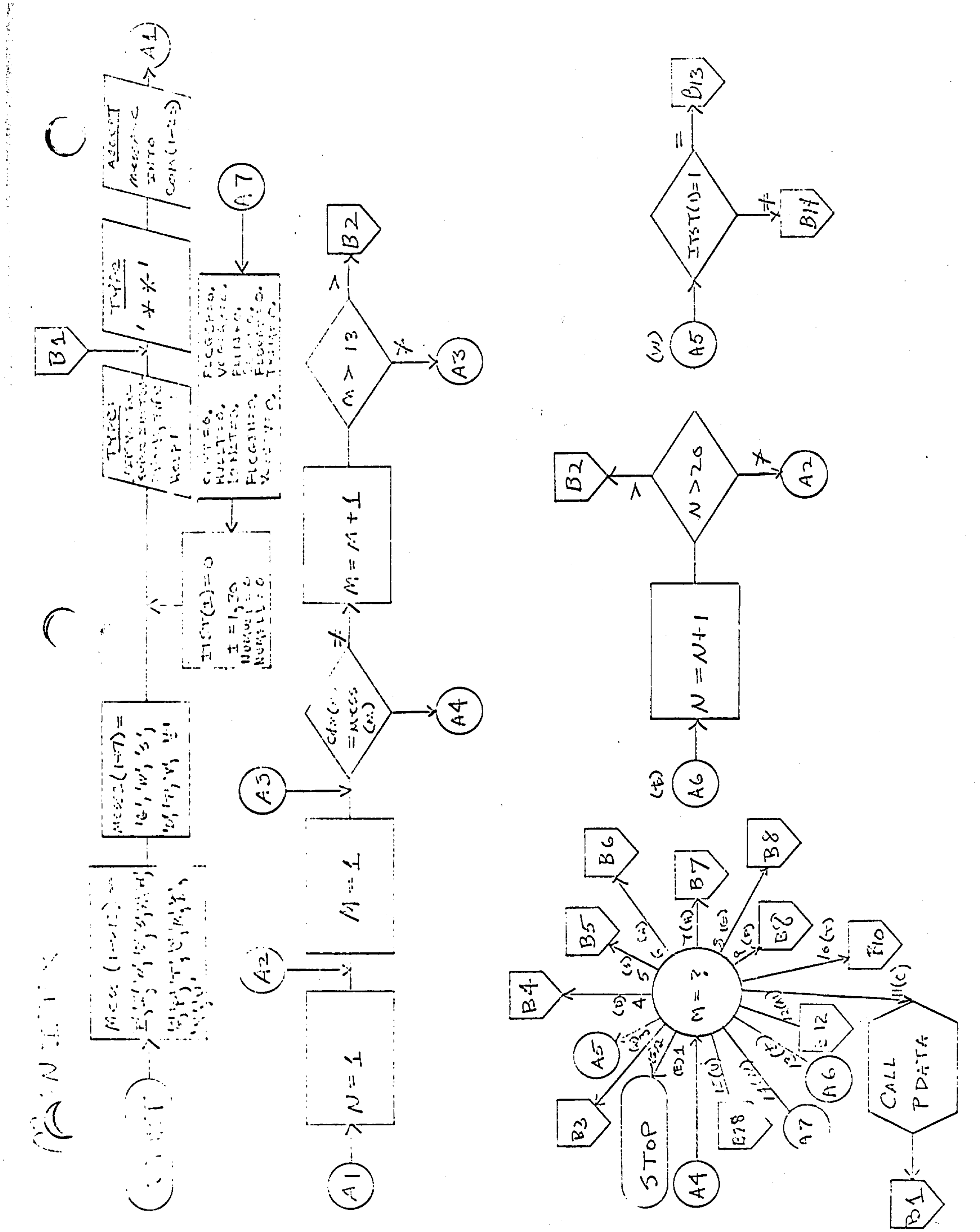
9) "NO DATA TO PLOT" - User tried to plot the motion of the separated capsule or hull (GRAPH}) but has not computed the capsule or hull motions. Corrective action: use the ASCENT} or HULL} command and try GRAPH} again.

10) "SORRY--CAPSULE HAS SUNK" - The user has tried to inspect the capsule's behavior at the surface (MOTION}) but the capsule has been found not buoyant. Corrective action: user should input new geometric characteristics and/or a new weight distribution (INPUT WEIGHTS}), (INPUT GEOMETRY}), and/or specify new floodable volumes (INPUT VOLUMES}), and try again.

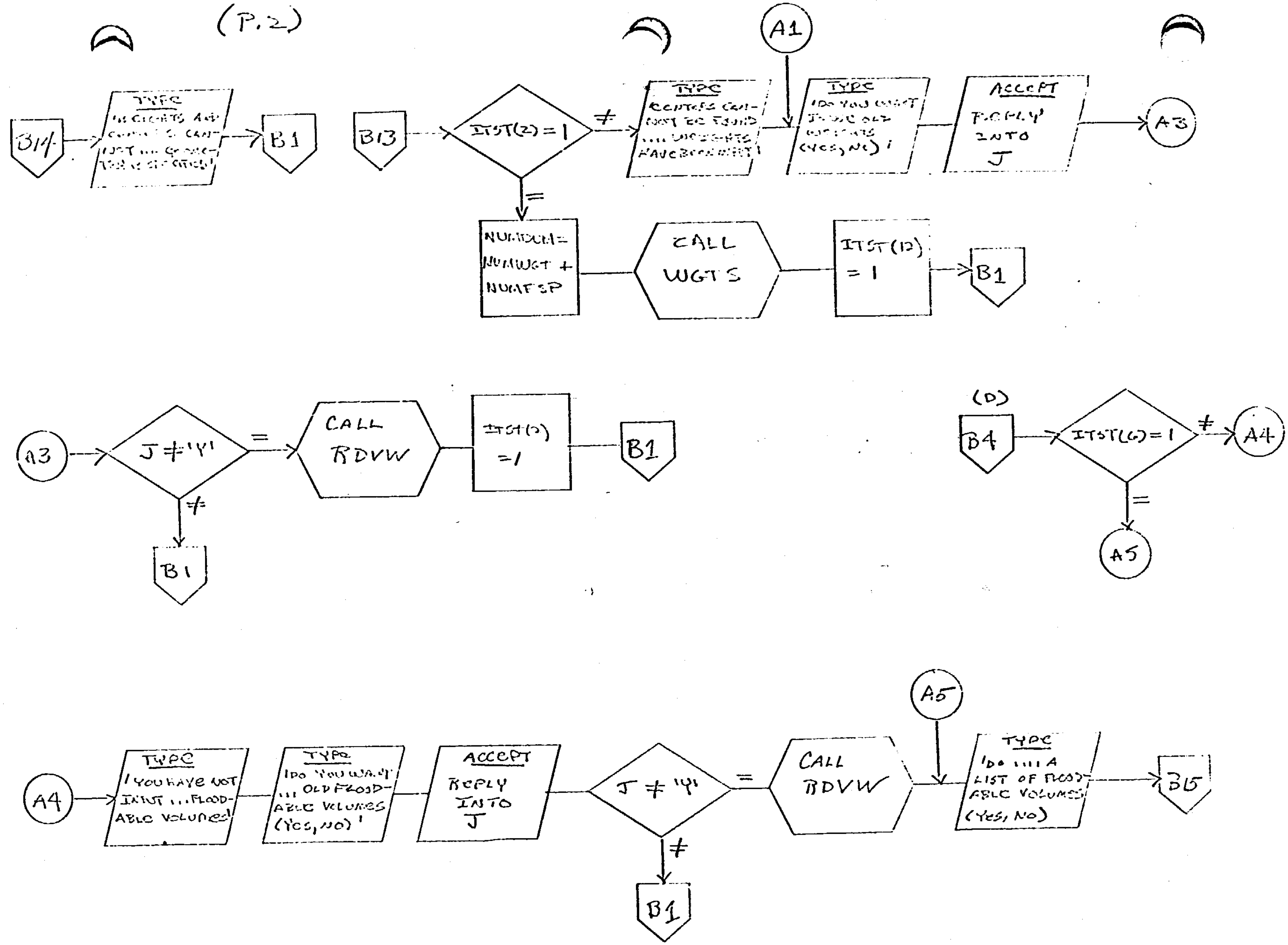
11) "NOT ENOUGH INPUT DATA TO CALCULATE SURFACE MOTIONS. INPUT SEA STATE" - User attempted to investigate capsule behavior on the surface (MOTIONS}) without specifying the sea state. Corrective action: input the desired surface sea state (INPUT SEAS}).

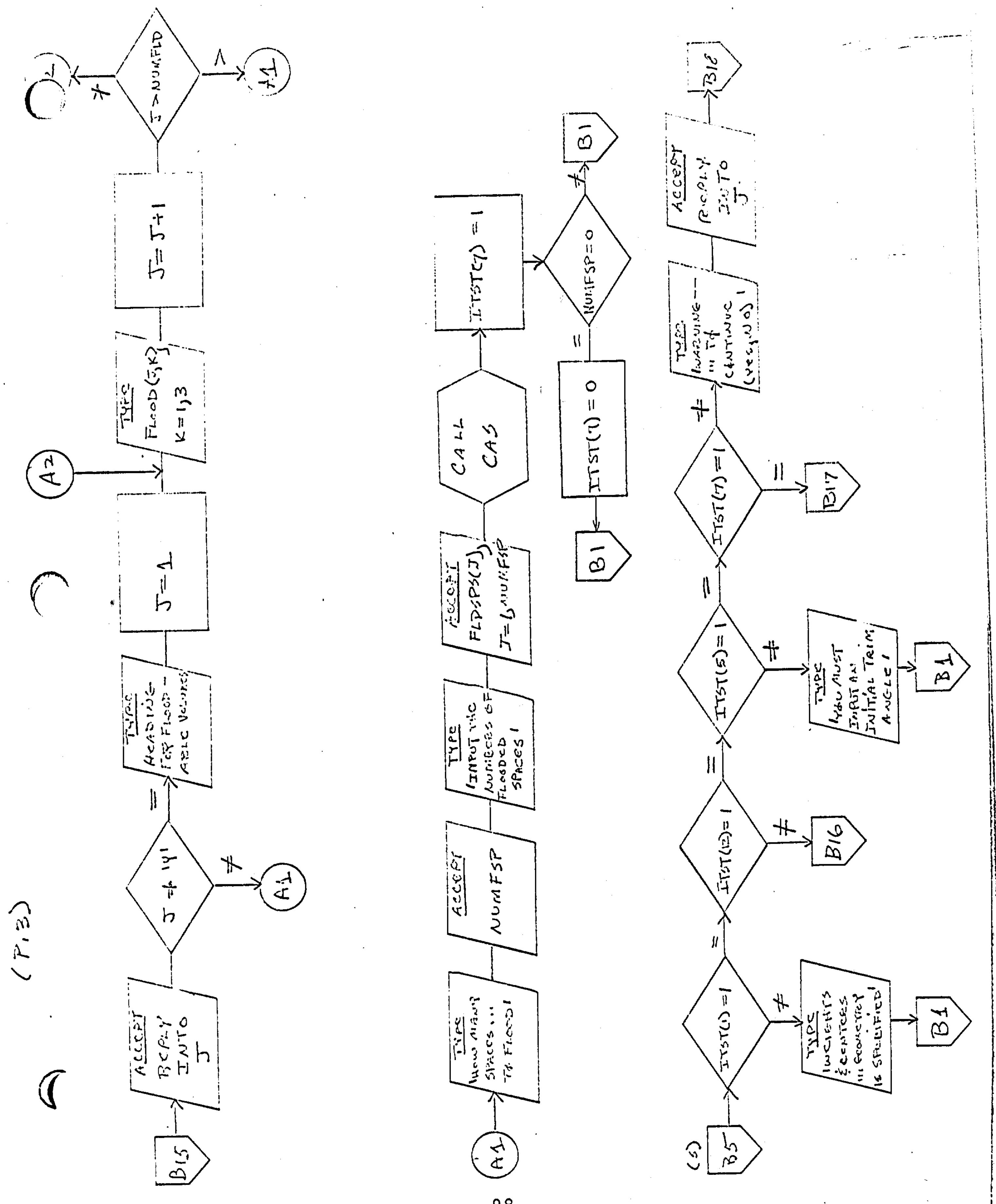
12) "ILLEGAL INPUT COMMAND, TRY AGAIN." - The user's command was not recognized by the monitor as a legal command. Corrective action: try again.

IX-1.1 Flow Chart for Main Program (MØNITR)

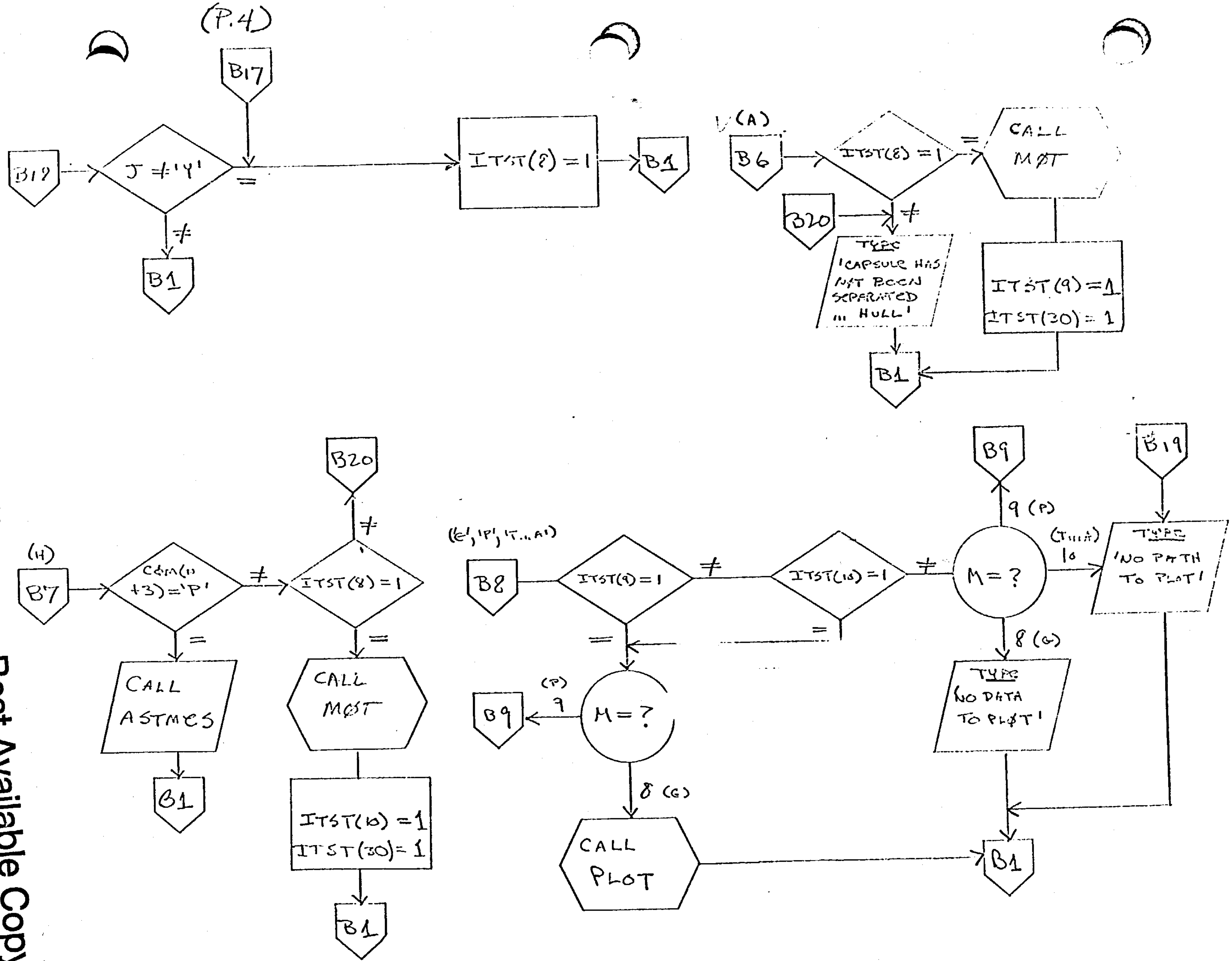

N = 1
M = 1
M = M + 1
M > 13
N = N+1
N > 20
ITST(I)=1
M = ?
STOP
CALL PDATA

(P.2)

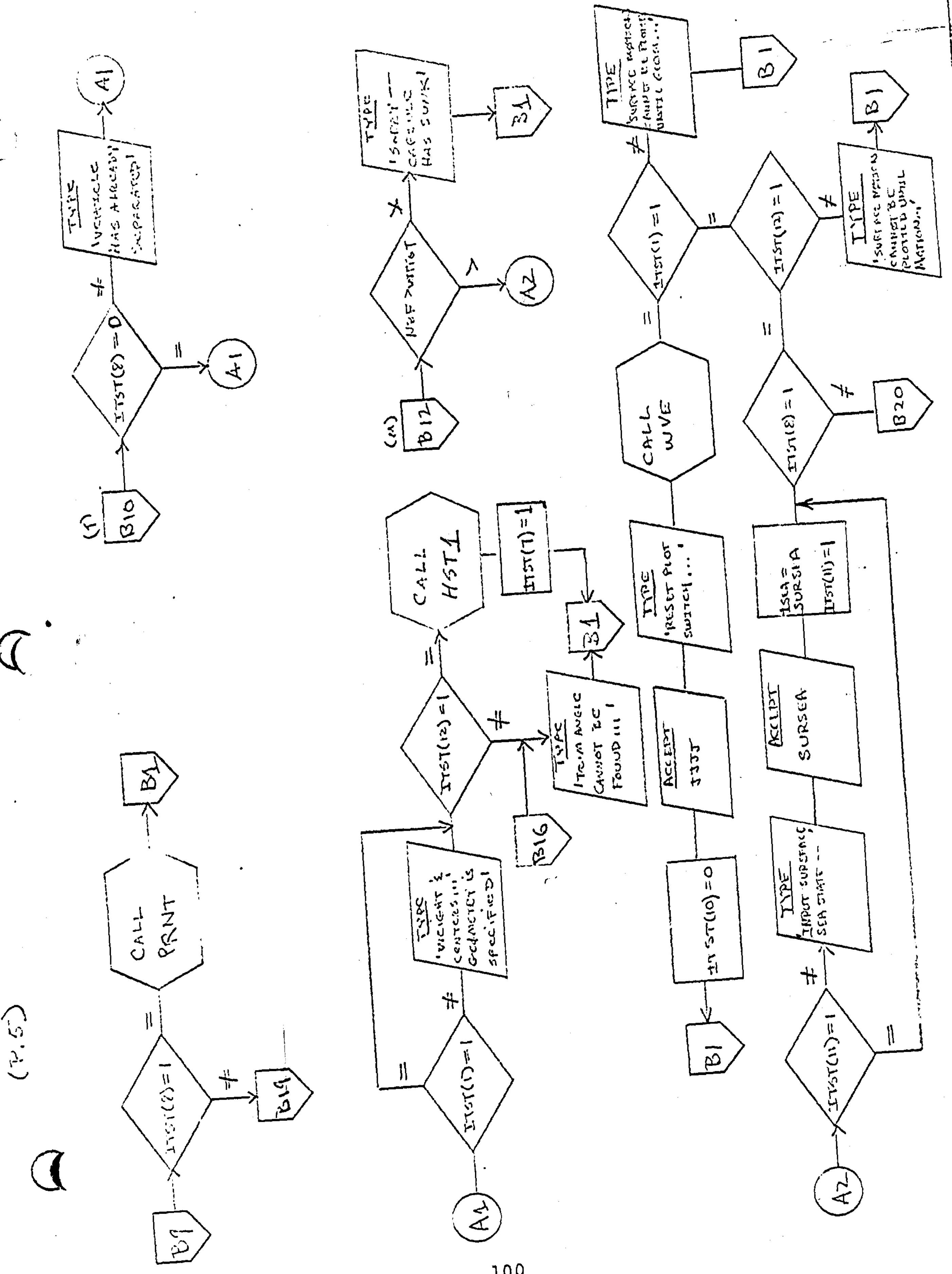

B14
TYPE
B1
B13
ITST(2) = 1
≠
=
TYPE
A1
TYPE
(YES, NO)
ACCEPT
REPLY INTO J
A3
NUMDUM = NUMWGT + NUMFSP
CALL WGTS
ITST(12) = 1
B1
A3
J ≠ 'Y'
=
≠
CALL RDVW
ITST(7) = 1
B1
B1
(D)
B4
ITST(6) = 1
≠
=
A4
A5
A4
TYPE
'YOU HAVE NOT INPUT ... FLOODABLE VOLUMES'
TYPE
'DO YOU WANT ... OLD FLOODABLE VOLUMES (YES, NO)'
ACCEPT
REPLY INTO J
J ≠ 'Y'
=
≠
B1
CALL RDVW
A5
TYPE
'DO ... A LIST OF FLOODABLE VOLUMES (YES, NO)'
B5

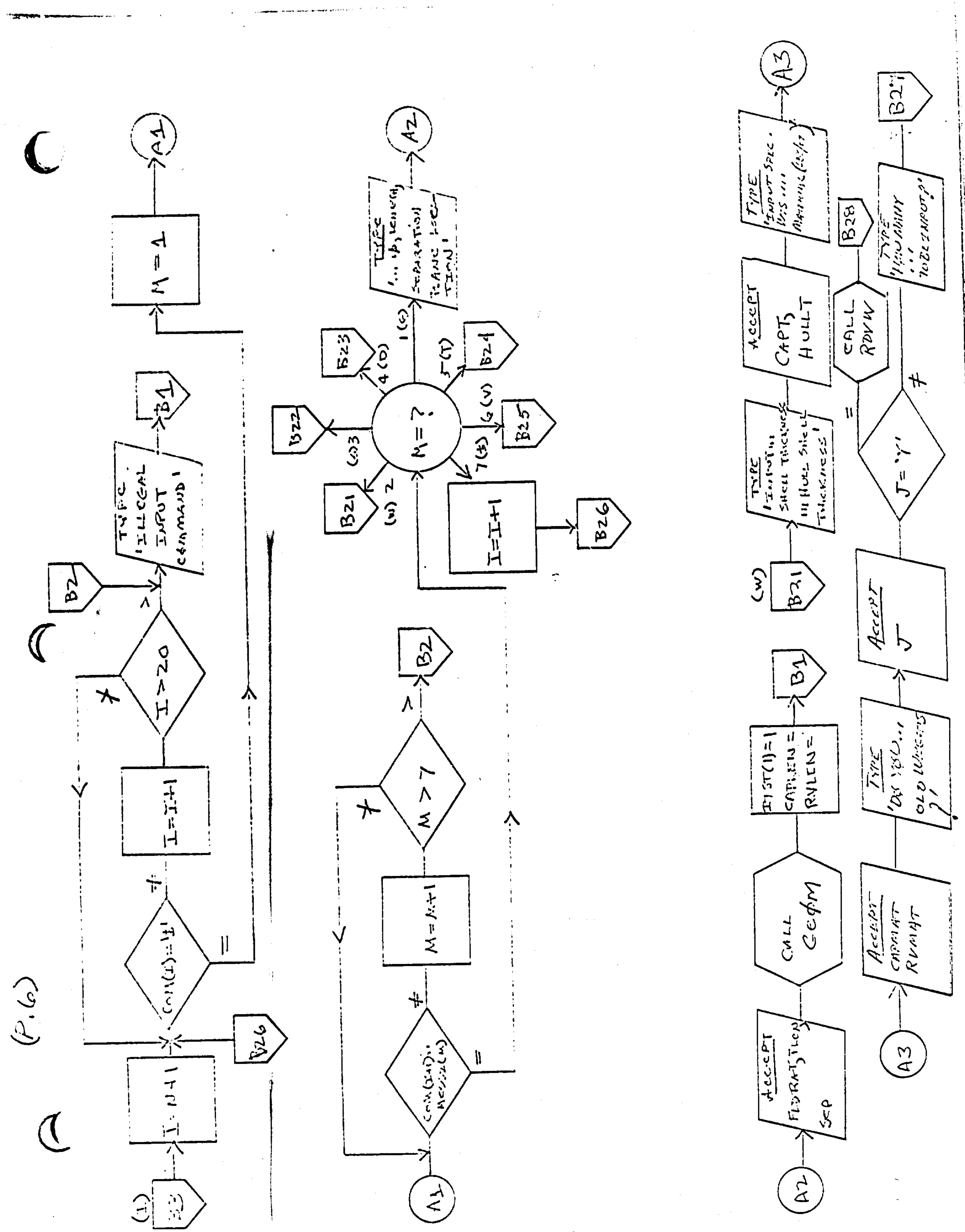


(P.3)
B15
ACCEPT REPLY INTO J
J ≠ 'Y'
≠
=
A1
TYPE HEADING FOR FLOOD-ABLE VOLUMES
J = 1
A2
TYPE FLOOD(J,K) K=1,3
J = J+1
J > NUMFLD
≠
>
A1
A1
TYPE 'HOW MANY SPACES ... TO FLOOD'
ACCEPT NUMFSP
TYPE 'INPUT THE NUMBERS OF FLOODED SPACES'
ACCEPT FLDSPS(J), J=1,NUMFSP
CALL CAS
ITST(7) = 1
NUMFSP=0
≠
B1
=
ITST(7) = 0
B1
(5)
B5
ITST(1) = 1
≠
TYPE 'WEIGHTS & CENTERS ... CORRECTED ... IS SPECIFIED'
B1
=
ITST(12) = 1
≠
B16
=
ITST(5) = 1
≠
TYPE 'YOU MUST INPUT AN INITIAL TRIM ANGLE'
B1
=
ITST(7) = 1
=
B17
≠
TYPE 'WARNING --- ... TO CONTINUE (YES,NO)'
ACCEPT REPLY INTO J
B18



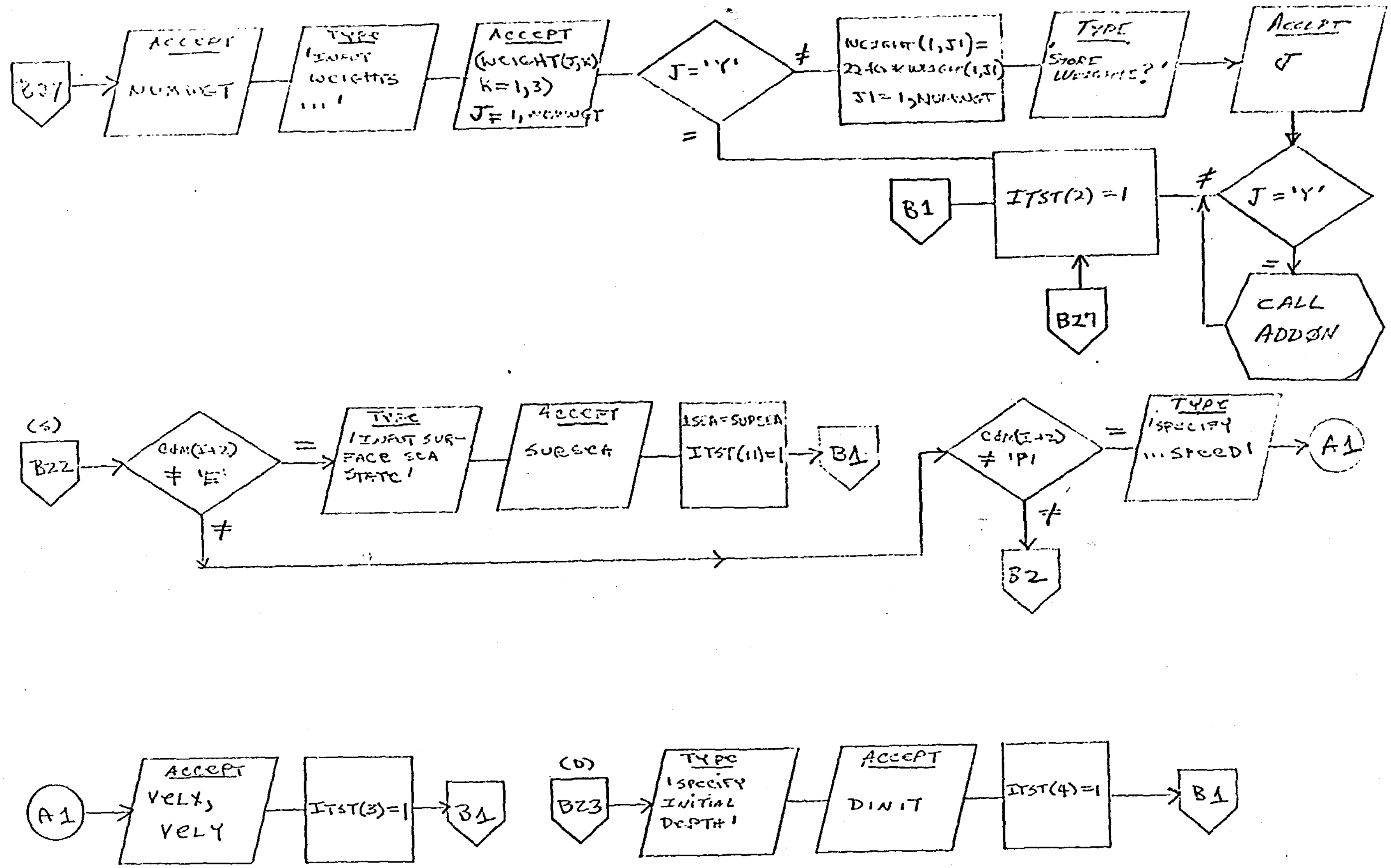
(P.4)
B17
B19
J ≠ 'Y'
=
≠
B1
ITST(8) = 1
B1
(A)
B6
ITST(8) = 1
=
CALL MØT
B20
≠
TYPE 'CAPSULE HAS NOT BEEN SEPERATED ... HULL'
B1
ITST(9) = 1
ITST(30) = 1
B1
B20
≠
(H)
B7
COM(I+3) = 'P'
≠
=
CALL ASTMES
B1
ITST(8) = 1
=
CALL MØST
ITST(10) = 1
ITST(30) = 1
B1
B8
ITST(9) = 1
≠
=
ITST(10) = 1
≠
=
M = ?
9 (P)
B9
10
TYPE 'NO PATH TO PLOT'
B19
8 (G)
TYPE 'NO DATA TO PLØT'
B1
M = ?
(P) 9
B9
8 (G)
CALL PLOT


(P.5)
B7
ITST(8)=1
CALL PRNT
B1
B19
(T)
B10
ITST(8)=0
TYPE 'VEHICLE HAS ALREADY SEPARATED!
A1
A1
A1
ITST(1)=1
TYPE 'WEIGHT & CENTERS!!!' GEOMETRY IS SPECIFIED!
ITST(12)=1
CALL HST1
ITST(7)=1
B16
TYPE 'TRIM ANGLE CANNOT BE FOUND!!! '
B1
(M)
B12
TYPE 'SAFETY — CAPSULE HAS SUNK!
A2
B1
CALL WVE
ITST(1)=1
TYPE 'SURFACE MOTION CANNOT BE PLOTTED UNTIL GEOM...'
B1
TYPE 'RESET PLOT SWITCH...'
ACCEPT JJJJ
ITST(10)=0
B1
A2
ITST(11)=1
TYPE 'INPUT SURFACE SEA STATE --'
ACCEPT SURSEA
ISEA = SURSEA ITST(11)=1
ITST(8)=1
B20
ITST(12)=1
TYPE 'SURFACE MOTION CANNOT BE PLOTTED UNTIL MOTION...'
B1

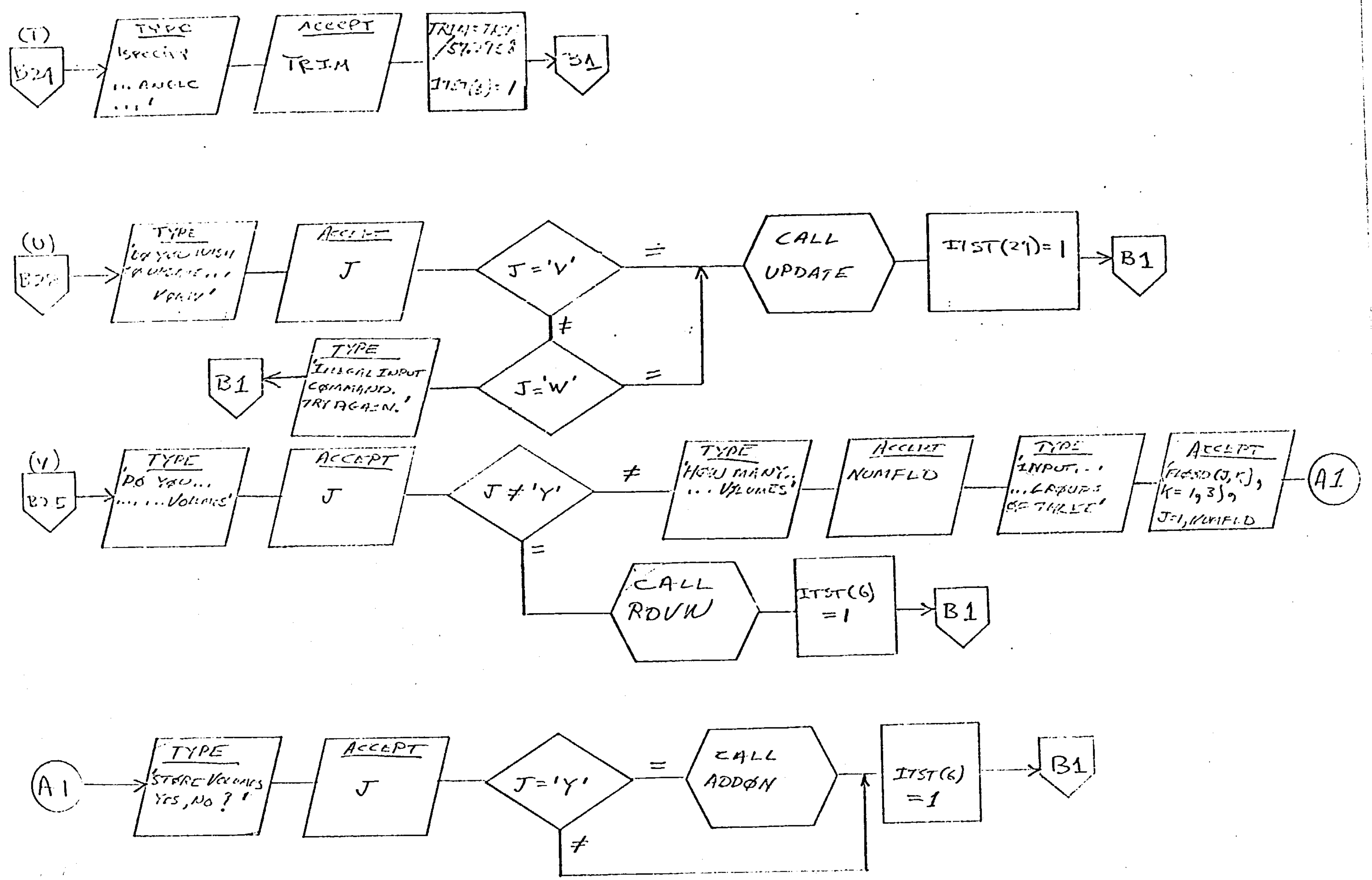


(P.6)
I = N+1
I = I+1
I > 20
M = 1
A1
B1
B2
B26
TYPE
'ILLEGAL INPUT COMMAND'
M = M+1
M > 7
M = ?
B21
B22
B23
B24
B25
A2
ACCEPT
CAPT, HULLT
CALL GEOM
CALL RDVW
B28
ACCEPT J
J = 'Y'
A3
ACCEPT
CAPMAT
RVMAT

(P.7)


B27
ACCEPT NUMWGT
TYPE 'INPUT WEIGHTS ...'
ACCEPT (WEIGHT(J,K) K=1,3) J=1,NUMWGT
J='Y'
≠
=
WEIGHT(1,J1)= 2240*WEIGHT(1,J1) J1=1,NUMWGT
TYPE 'STORE WEIGHTS?'
ACCEPT J
J='Y'
CALL ADDON
ITST(2)=1
B1
B27
(S)
B22
CDM(I+2) ≠ 'E'
TYPE 'INPUT SURFACE SEA STATE'
ACCEPT SURSEA
ISEA=SURSEA ITST(11)=1
B1
CDM(I+2) ≠ 'P'
TYPE 'SPECIFY ...SPEED'
A1
B2
A1
ACCEPT VELX, VELY
ITST(3)=1
B1
(D)
B23
TYPE 'SPECIFY INITIAL DEPTH'
ACCEPT DINIT
ITST(4)=1
B1

(P. 8)


(T)
B24
TYPE
ACCEPT
TRIM
ITST(8) = 1
B1
(U)
TYPE
ACCEPT
J
J = 'V'
J = 'W'
CALL
UPDATE
IIST(27) = 1
B1
TYPE
'ILLEGAL INPUT COMMAND, TRY AGAIN.'
B1
(V)
TYPE
'DO YOU... ...VOLUMES'
ACCEPT
J
J ≠ 'Y'
TYPE
'HOW MANY... ...VOLUMES'
ACCEPT
NUMFLD
TYPE
'INPUT... ...GROUPS OF THREE'
ACCEPT
(FLDSD(J,K), K = 1,3), J=1,NUMFLD
A1
CALL
RDVW
ITST(6) = 1
B1
A1
TYPE
'STORE VOLUMES YES, NO ?'
ACCEPT
J
J = 'Y'
CALL
ADDON
ITST(6) = 1
B1

IX-1.2 Listing of Main Program (MØNITR)

```
      COMMON/INPUT/FLEN,SEP,FLDRAT,TRIM,VELX,VELY,VELL
      COMMON/INOUT/FLCBIN,FLCBCP,FLCBRV,CGCAP1,CGRV1,VOLITV,ELCBIN,
     * SITV,VOLCAP,ELCBCP,SCAP,RGCAP,VOLRV,SRV,RGRV,BAREA,PROJCP,
     * PROJRV,TOTWT,RVWT,CAPWT,FLCGIN,VCGITV,FLCGRV,VCGRV,FLCGCP,
     * VCGCAP,TIME(101),ANGL(101),VDIS(101),HDIS(101),
     * AVEL(101),VVEL(101),HVEL(101)
      INTEGER COM
      DIMENSION IFLDSP(10)
      DIMENSION ITST(30),COM(20),MESS(15),MESS2(7),FLOOD(3,
     1 100),WEIGHT(3,100)
      DATA (MESS(I),I=1,15),(MESS2(J),J=1,7)/'E','I','W','D',
     1 'S','A','H','G','P','T','C','M',' ','N','U','G','W','S','D'
     2 ,'T','V',' '/
      CALL ERRSET(0)
      NUMFSP=0
      GO TO 2008
5     DO 10 I=1,30
 10   ITST(I)=0
      NUMWGT=0
      NUMFLD=0
      CAPT=0.
      HULLT=0.
      DINIT=0.
      FLCGIN=0.
      VCGITV=0.
      FLCGCP=0.
      VCGCAP=0.
      FLEN=0.
      SEP=0.
      FLDRAT=0.
      TRIM=0.
2008  TYPE 15
15    FORMAT(' IF YOU REQUIRE INSTRUCTIONS, TYPE "HELP".')
 20   TYPE 25
 25    FORMAT(1H ,'**',$)
      ACCEPT 30,(COM(I),I=1,20)
 30   FORMAT(20A1)
      DO 660 N=1,20
1100  DO 1000 M=1,15
      IF(COM(N).EQ.MESS(M)) GO TO 1500
1000  CONTINUE
      GO TO 711
1500  GOTO(9999,700,600,610,620,625,630,635,635,615,650,
     1 655,660,5,2000),M
9999  STOP
 660  CONTINUE
      GO TO 711
 600  IF(ITST(1)-1) 6001,6005,6001
6001  TYPE 6002
6002  FORMAT(' WEIGHTS AND CENTERS CANNOT BE FOUND UNTIL'
     1 ,' GEOMETRY IS SPECIFIED')
      GO TO 20
6005  IF(ITST(2)-1) 6006,6008,6006
```

```
6006   TYPE 6007
6007   FORMAT(' CENTERS CANNOT BE FOUND UNTIL THE WEIGHTS '
   1   'HAVE BEEN INPUT')
6010  TYPE 6011
6011   FORMAT(' DO YOU WANT TO USE OLD WEIGHTS? (YES OR NO)',' -- ',$)
       ACCEPT 30, J
       IF(J.NE.'Y') GO TO 20
       CALL RDVW(WEIGHT,IDENT,NUMWGT,0)
       ITST(2)=1
       GO TO 20
6008   NUMDUM=NUMWGT+NUMFSP
       CALL WGTS(NUMDUM,WEIGHT,CAPT,HULLT,CAPMAT,RVMAT,
   *   CINERT,HINERT)
       ITST(12)=1
       GO TO 20
 610   IF(ITST(6)-1) 6110,611,6110
6110   TYPE 6111
6111   FORMAT(' YOU HAVE NOT INPUT A LIST OF FLOODABLE VOLUMES')
       TYPE 6112
6112   FORMAT(' DO YOU WANT TO USE OLD FLOODABLE VOLUMES? (YES'
   1   ' OR NO)',' -- ',$)
       ACCEPT 30, J
       IF(J.NE.'Y') GO TO 20
       CALL RDVW (FLOOD,IDENT,NUMFLD,1)
 611   TYPE 6100
6100   FORMAT(' DO YOU NEED A LIST OF FLOODABLE VOLUMES? (YES'
   1   ' OR NO)',' -- ',$)
       ACCEPT 30, J
       IF(J.NE.'Y') GOTO 6106
       TYPE 6107
6107   FORMAT(' SPACE NO',5X,'VOL (CU.FT.)',5X,'LONG CG',5X,
   *   'VERT CG',/,' --------',5X,'------------',5X,
   *   '-------',5X,'-------',/)
       DO 6108 J=1,NUMFLD
6108   TYPE 6101,J,(FLOOD(K,J),K=1,3)
6101   FORMAT(1H ,3X,I2,10X,F8.1,7X,F7.3,5X,F7.3)
6106   TYPE 6102
6102   FORMAT('OHOW MANY SPACES DO YOU WANT TO FLOOD?',' -- ',$)
       ACCEPT 7203,NUMFSP
       TYPE 6103
6103   FORMAT(' INPUT THE NUMBERS OF FLOODED SPACES'/)
       ACCEPT 6104,(IFLDSP(J),J=1,NUMFSP)
6104   FORMAT(10I)
       CALL CAS(WEIGHT,NUMWGT,FLOOD,NUMFLD,IFLDSP,NUMFSP)
6105   ITST(7)=1
       IF(NUMFSP.EQ.0)ITST(7)=0
       GO TO 20
615    CONTINUE
```

```
      IF(ITST(8)) 6150,6153,6150
6150  TYPE 6151
6151 FORMAT (1H ,'VEHICLE HAS ALREADY SEPARATED')
6153  IF(ITST(1)-1) 6152,6156,6152
6152  TYPE 6002
6156   IF(ITST(12)-1) 6157,6159,6157
6157  TYPE 6158
6158  FORMAT(' TRIM ANGLE CANNOT BE FOUND UNTIL WEIGHTS AND'
    1 ' CENTERS ARE FOUND')
      GO TO 20
6159  CALL HST1(ITST(7))
      GO TO 20
 620  IF(ITST(1)-1) 6200,6202,6200
6200   TYPE 6002
      GO TO 20
6202  IF(ITST(12)-1) 6157,6204,6157
6204  IF(ITST(5)-1) 6205,6207,6205
6205  TYPE 6206
6206  FORMAT(' YOU MUST INPUT AN INITIAL'
    1 ' TRIM ANGLE')
      GOTO 20
6207  IF(ITST(7)-1) 6208,6210,6208
6208  TYPE 6209
6209  FORMAT(' WARNING--NO CASULTY SPECIFIED. DO YOU WANT TO'
    1 ' CONTINUE? (YES OR NO)',' -- ',$)
      ACCEPT 30,J
      IF(J.NE.'Y') GO TO 20
6210  CONTINUE
      ITST(8)=1
      GO TO 20
625   IF(ITST(8)-1)6250,6252,6250
6250  TYPE 6251
6251  FORMAT(' CAPSULE HAS NOT BEEN SEPARATED FROM MAIN HULL')
      GO TO 20
6252  CALL MOT(CAPLEN,CAPWT,VOLCAP,CINERT,RGCAP,SCAP,FLCGCP,
    * VCGCAP,FLCBCP,PROJCP,NPTS)
      FL=CAPLEN
      ITST(9)=1
      ITST(30)=1
      GO TO 20
 630  IF(COM(N+3).EQ.'P') GO TO 9500
      IF(ITST(8)-1) 6250,6300,6250
6300  CALL MOT(RVLEN,RVWT,VOLRV,RINERT,RGRV,SRV,FLCGRV,
    * VCGRV,FLCBRV,PROJRV,NPTS)
      FL=RVLEN
      ITST(10)=1
      ITST(30)=2
      GO TO 20
```

```
635    IF(ITST(9)-1)6350,6352,6350
6350   IF(ITST(10)-1) 6353,6352,6353
6352   GO TO (6355,6401),M-7
6353   GO TO (6351,6401,6451),M-7
6351   TYPE 6354
6354   FORMAT(' NO DATA TO PLOT')
       GO TO 20
6355   CALL PLOT(ITST(30),NPTS)
       TYPE 2009
2009   FORMAT(' RESET PLOT SWITCH TO TEXT MODE AND',
     *' GIVE A CARRIAGE RETURN -- ',$)
       ACCEPT 30,JJJJ
       ITST(10)=0
       GO TO 20
6401   IF(ITST(8)-1) 6451,6405,6451
6405   CALL PRNT(FL,NPTS)
       GO TO 20
6451   TYPE 6452
6452   FORMAT(' NO PATH TO PLOT')
       GO TO 20
 650   CALL PDATA(ITST,FLOOD,WEIGHT,NUMWGT,NUMFLD,CAPT,HULLT,
     * DINIT,FLCGIN,VCGITV,FLCGCP,VCGCAP)
       GOTO 20
 655   IF(VOLCAP*64.-CAPWT) 6550,6550,6552
6550   TYPE 6551
6551   FORMAT(' SORRY--CAPSULE HAS SUNK')
       GO TO 20
6552   IF(ITST(11)-1) 6553,6555,6553
6553   TYPE 7250
       ACCEPT 7151,SURSEA
       ISEA=SURSEA
       ITST(11)=1
6555   IF(ITST(8)-1) 6250,6556,6250
6556   IF(ITST(12)-1)6557,6558,6557
6557   TYPE 6559
6559   FORMAT(' SURFACE MOTION CANNOT BE PLOTTED UNTIL WEIGHTS'/
     * ' AND CENTERS ARE FOUND.'/)
       GO TO 20
6558   IF(ITST(1)-1)6560,6561,6560
6560   TYPE 6562
6562   FORMAT(' SURFACE MOTION CANNOT BE PLOTTED UNTIL GEOMETRY'/
     * ' IS SPECIFIED.'/)
       GO TO 20
6561   CALL WVE(FLDRAT,VCGCAP,FLEN,BAREA,CAPLEN,CAPWT,FLCGCP,ISEA)
       TYPE 2009
       ACCEPT 30,JJJJ
       GO TO 20
```

```
700     DO 710 I=N+1,20
7100    IF(COM(I).EQ.' ') GO TO 7110
710     CONTINUE
711     TYPE 35
35      FORMAT(' ILLEGAL INPUT COMMAND, TRY AGAIN.')
        GO TO 20
7110    DO 7111 M=1,7
        IF(COM(I+1).EQ.MESS2(M)) GO TO 7112
7111    CONTINUE
        GO TO 711
7112    GOTO(715,720,725,730,735,740,745),M
715     TYPE 7150
7150    FORMAT(' SPECIFY L/D, LENGTH, SEPARATION PLANE -- ',$)
        ACCEPT 7151,FLDRAT,FLEN,SEP
7151     FORMAT(3F)
        CALL GEOM
        ITST(1)=1
        CAPLEN=FLEN*SEP
        RVLEN=FLEN-CAPLEN
        GO TO 20
720     TYPE 7200
7200    FORMAT(' INPUT AVERAGE CAPSULE SHELL THICKNESS (IN),'
     * ' AVERAGE',/,' HULL SHELL THICKNESS (IN) -- ',$)
        ACCEPT 7151,CAPT,HULLT
        TYPE 7201
7201    FORMAT(' INPUT SPECIFIC WEIGHTS OF CAPSULE MATERIAL, AND',/,
     * ' OF HULL MATERIAL (LBS/FT**3) --  ',$)
        ACCEPT 7151,CAPMAT,RVMAT
        TYPE 2002
2002    FORMAT(' DO YOU WANT TO USE OLD WEIGHTS?'/
     *  ' YES OR NO -- '$)
        ACCEPT 30,J
        IF(J.NE.'Y')GO TO 2003
        CALL RDVW(WEIGHT,IDENT,NUMWGT,0)
        GO TO 2004
2003        TYPE 7202
7202    FORMAT(' HOW MANY WEIGHTS ARE TO BE INPUT?',' -- ',$)
        ACCEPT 7203,NUMWGT
7203    FORMAT(I)
        TYPE 7204
7204    FORMAT(' INPUT WEIGHTS IN LONG TONS, LONG"L CG, VERT CG'
     1  /' IN GROUPS OF THREE'/)
        ACCEPT 7151,((WEIGHT(K,J1),K=1,3),J1=1,NUMWGT)
        IF(J.EQ.'Y')GO TO 2004
2005    DO 7210 J1=1,NUMWGT
7210    WEIGHT(1,J1)=2240.*WEIGHT(1,J1)
        TYPE 7205
7205    FORMAT(' STORE WEIGHTS? (YES OR NO)',' -- ',$)
        ACCEPT 30,J
        IF(J.EQ.'Y') CALL ADDON(WEIGHT,0,21,NUMWGT)
2004        ITST(2)=1
        GO TO 20
```

```
 725   IF(COM(I+2).NE.'E') GO TO 7255
 726   TYPE 7250
7250   FORMAT(' INPUT SURFACE SEA STATE',' -- ',$)
       ACCEPT 7151, SURSEA
       ISEA=SURSEA
       ITST(11)=1
       GO TO 20
7255   IF(COM(I+2).NE.'P') GO TO 711
       TYPE 7256
7256   FORMAT(' SPECIFY INITIAL VERTICAL AND HORIZONTAL '
     1 'SPEED (FT/SEC) -- ',$)
       ACCEPT 7151,VELY,VELX
       ITST(3)=1
       GO TO 20
 730   TYPE 7301
7301   FORMAT(' SPECIFY INITIAL DEPTH (FT)',' -- ',$)
       ACCEPT 7151,DINIT
       ITST(4)=1
       GO TO 20
 735   TYPE 7350
7350   FORMAT(' SPECIFY INITIAL TRIM ANGLE (DEGREES)',' -- ',$)
       ACCEPT 7151, TRIM
       TRIM=TRIM/57.2958
       ITST(5)=1
       GO TO 20
 740   TYPE 2006
2006   FORMAT(' DO YOU WANT TO USE OLD FLOODABLE VOLUMES?'/
     * ' YES OR NO -- '$)
       ACCEPT 30,J
       IF(J.NE.'Y')GO TO 2007
       CALL RDVM(FLOOD,IDENT,NUMFLD,1)
       ITST(6)=1
       GO TO 20
2007       TYPE 7400
7400   FORMAT(' HOW MANY FLOODABLE VOLUMES',' -- ',$)
       ACCEPT 7203,NUMFLD
       TYPE 7401
7401   FORMAT(' INPUT FLOODABLE VOLUMES, LONG"L, AND VERTICAL'
     1 ' CENTROIDS'/' IN GROUPS OF THREE'/)
       ACCEPT 7151, ((FLOOD(K,J),K=1,3),J=1,NUMFLD)
       TYPE 7402
7402   FORMAT(' STORE VOLUMES? (YES OR NO)',' -- ',$)
       ACCEPT 30, J
       IF(J.EQ.'Y') CALL ADDON(FLOOD,1,23,NUMFLD)
       ITST(6)=1
       GO TO 20
```

```
 745    I=I+1
        GO TO 7100
2000    TYPE 2001
2001    FORMAT(' DO YOU WISH TO UPDATE WEIGHTS OR VOLUMES?'/
     *  ' TYPE V OR W  -- ',$)
        ACCEPT 30,J
        IF(J.EQ.'V')GO TO 1999
        IF(J.EQ.'W')GO TO 1998
        GOTO 711
1999    CALL UPDATE(FLOOD,1,NUMFLD)
        ITST(20)=1
        GOTO 20
1998    CALL UPDATE(WEIGHT,0,NUMWGT)
        ITST(20)=1
        GO TO 20
9500    CALL ASTMES
        GOTO 20
        END
```

IX-2. Subroutine GEØM

The volume, surface area, volume moment of inertia, radius of gyration, longitudinal center of bouyancy, and the longitudinal center of gravity of the shell are calculated in the subroutine GEØM. The subroutine accepts the values of the separation point, length-to-diameter ratio and vehicle length. The non-dimensional variables defining the geometry of the intact vehicle [R1=0, S1=2] are set and the program passed to the basic geometry calculation routine where non-dimensional values for the required parameters are calculated. The program is then returned to Statement 1, where these values for the intact vehicle are saved. The non-dimensional variables defining the capsule [R1=0, S1= the separation point] are then set and the program is returned to Statement 2 where these values for the capsule are saved. The non-dimension variables defining the remaining vehicle [R1 = separation point, S1 = 2.0] are set and the program again passed to the basic geometry calculation routine. When the required parameters for the remaining vehicle have been calculated the program then passes to Statement 30 where the required dimensional parameters are calculated. The program then returns to the monitor.

This subroutine contains no error messages.

Input Variables

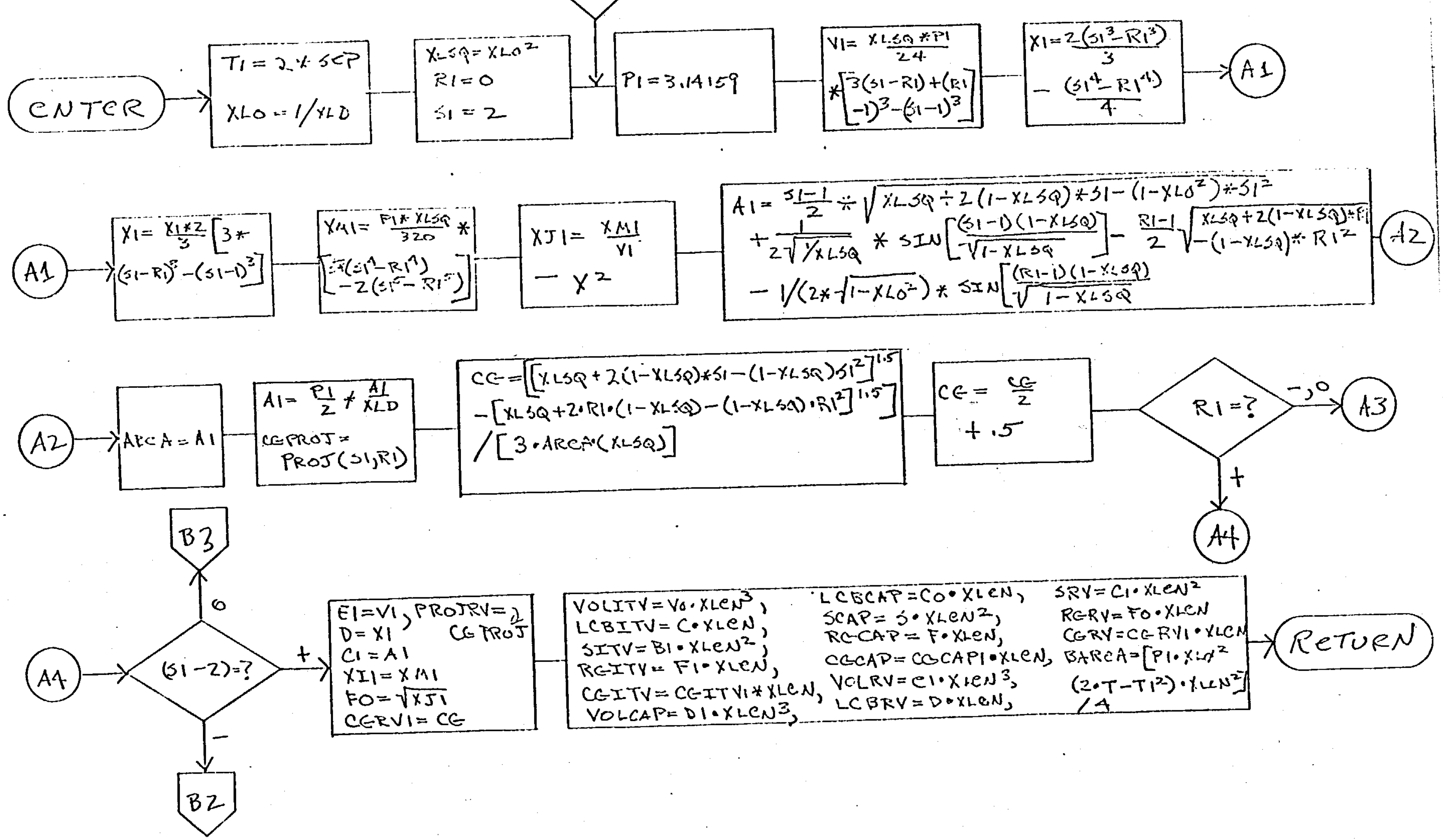
XLEN	Length of intact vehicle (ft.)
SEP	Separation point from nose (decimal fraction)
XLD	Length to diameter ratio

Output Variables

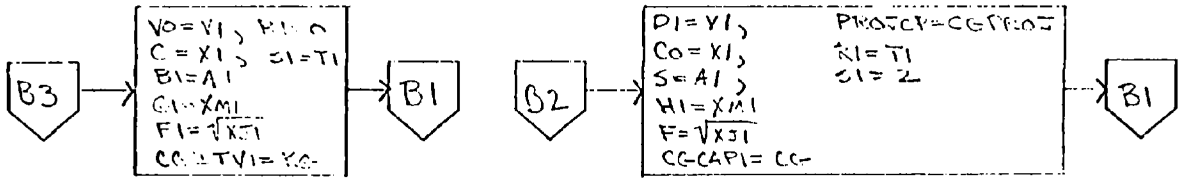
C	Longitudinal center of buoyancy/XLEN, intact vehicle
CØ	Longitudinal center of buoyancy/XLEN, capsule
D	Longitudinal center of buoyancy/XLEN, remaining vehicle
CGCAP1	Center of gravity/XLEN, capsule shell
CGRV1	Center of gravity/XLEN, remaining vehicle shell
VØLITV	Volume of intact vehicle (ft.^{3})
LCBITV	Longitudinal center of buoyancy of intact vehicle (ft. from nose)
SITV	Surface area of intact vehicle (ft.^{2})
VØLCAP	Volume of capsule (ft.^{3})
LCBCAP	Longitudinal center of buoyancy of capsule (ft. from nose)
SCAP	Surface area of capsule (ft.^{2})
RGCAP	Radius of gyration of capsule (ft.)
VOLRV	Volume of remaining vehicle (ft.^{3})
SRV	Surface area of remaining vehicle (ft.^{2})
RGRV	Radius of gyration of remaining vehicle (ft.)
BAREA	Base area of capsule and remaining vehicle (ft.^{2})
PRØJCP	Center of pressure of projected capsule area/XLEN
PRØJRV	Center of pressure of projected remaining vehicle area/XLEN

IX-2.1 Flow Chart for Subroutine GEØM

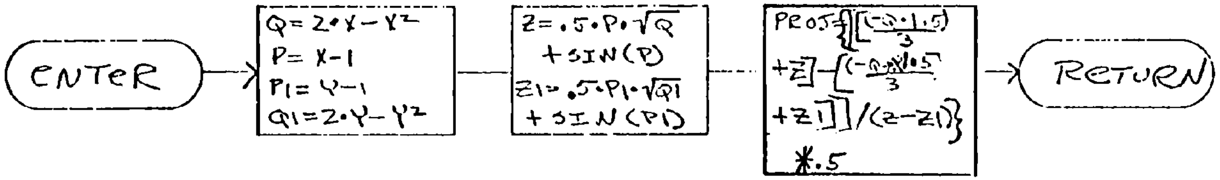
SUBROUTINE GEOM



SUBROUTINE GEOM (P.2)



FUNCTION TROJ (X, Y)



IX-2.2 Listing of Subroutine GEØM

```
      SUBROUTINE GEOM
      REAL LCBITV,LCBCAP,LCBRV
       COMMON/INPUT/XLEN,SEP,XLD,SKIP(4)
      COMMON/INOUT/C,CO,D,CGCAP1,CGRV1,VOLITV,LCBITV,SITV,
   *VOLCAP,LCBCAP,SCAP,RGCAP,VOLRV,SRV,RGRV,BAREA,PROJCP,
   *PROJRV,EXTRA(716)
      T1 = 2.*SEP
      XLO= 1/XLD
C
C     CALCULATION FOR INTACT VEHICLE
C
      XLSQ=XLO*XLO
      R1 = 0.0
       S1 = 2.0
      GO TO 10
    1 VO = V1
      C = X1
      B1 = A1
      G1 = XM1
      F1 = SQRT (XJ1)
      CGITV1 = CG
C
C      CALCULATION FOR CAPSULE
C
      R1 = 0
      S1 = T1
      GO TO 10
    2  D1 = V1
      CO = X1
      S = A1
      H1 = XM1
      F = SQRT (XJ1)
      CGCAP1 = CG
        PROJCP=CGPROJ
C
C     CALCULATION FOR REMAINRN4HUGHICLE
C
      R1 = T1
      S1 = 2.
      GO TO 10
    3 E1 = V1
      D = X1
       C1 = A1
      XI1 = XM1
       FO = SQRT (XJ1)
     CGRV1 = CG
       PROJRV=CGPROJ
       GO TO 30
```

```
C
C       BASIC GEOM. CALC. PACKAGE
C
   10   P1 = 3.14159
        V1 = (P1/24.)*XLSQ*(3.*(S1-R1)+(R1-1.)**3-(S1-1.)**3)
        X1 = (2./3.)*(S1**3-R1**3)-(1./4.)*(S1**4-R1**4)
         X1 = (X1)/((2./3.)*(3.*(S1-R1)+(R1-1.)**3-(S1-1.)**3))
        XM1= (P1/320.)*XLSQ*(5.*(S1**4-R1**4)-2.*(S1**5-R1
      1 **5))
        XJ1 = (XM1/V1) -X1*X1
        A1 = ((S1-1.)/2.)*SQRT(XLSQ+2.*(1.-XLSQ)*S1-(1.-XLO
      1 *XLO)*S1*S1)+(1./(2.*SQRT(1.-XLSQ)))*ASIN(((S1-1.)*(1.-
      2 XLSQ))/(SQRT(1.-XLSQ)))-((R1-1.)/2.)*SQRT(XLSQ+2.
      3 *(1.-XLSQ)*R1-(1.-XLSQ)*R1*R1)-(1./(2.*SQRT(1.-XLO*
      4 XLO)))*ASIN(((R1-1.)*(1.-XLSQ))/(SQRT(1.-XLSQ)))
        AREA=A1
        A1 = (P1/2.)*A1/XLD
C
         CGPROJ=PROJ(S1,R1)
C       CALCULATE C.G. OF SHELL
C
 25     CG=(((XLSQ+2.*(1.-XLSQ)*S1-(1.-XLSQ)*S1*S1)**1.5)
      * -((XLSQ+2.*(1.-XLSQ)*R1-(1.-XLSQ)*R1*R1)**1.5))
      * /(3.*(XLSQ-1.)*(AREA))
        CG=(CG/2.)+.5
        IF(R1)20,20,3
  20    IF(S1-2.) 2,1,3
C
 30     CONTINUE
C       OUTPUT
C
        VOLITV = VO*XLEN**3
        LCBITV = C*XLEN
        SITV = B1 *XLEN*XLEN
        RGITV =F1 * XLEN
        CGITV = CGITV1 * XLEN
        VOLCAP = D1 * XLEN **3
        LCBCAP = CO * XLEN
         SCAP = S*XLEN*XLEN
        RGCAP = F*XLEN
        CGCAP = CGCAP1*XLEN
         VOLRV = E1*XLEN**3
       LCBRV = D*XLEN
        SRV = C1*XLEN*XLEN
        RGRV = FO*XLEN
        CGRV = CGRV1*XLEN
         BAREA=(P1*(XLO)**2*(2.*T1-T1*T1)*XLEN*XLEN)/4.
        TYPE 500,VOLCAP,LCBCAP,CGCAP
 500    FORMAT(' CAPSULE VOLUME                      =',F12.2,/
      *        ' CAPSULE LONG. CENTER OF BOUYANCY  =',F10.3,/
      *        ' HYDRODYNAMIC CENTER OF CAPSULE    =',F10.3,//)
        RETURN
              END
```

```
 FUNCTION PROJ(X,Y)
  Q=2.*X-X*X
P=X-1.
P1=Y-1.
Q1=2.*Y-Y*Y
Z=.5*P*Q**.5+ASIN(P)
Z1=.5*P1*Q1**.5+ASIN(P1)
PROJ=(((-Q**1.5/3+Z)-(-Q1**1.5/3+Z1))/(Z-Z1))*.5
RETURN
 END
```

IX-3. Subroutine WGTS

As inputs the subroutine accepts the number of weights, the weight array, the separation point, the vehicle length, the length-to-diameter ratio, the thickness of the capsule and remaining vehicle shell material as well as the longitudinal center of gravity of the capsule and remaining vehicle shell.

The program then calculates the total weight of the intact vehicle, the capsule, and the remaining vehicle components. Subsequently the non-dimensional longitudinal and vertical centers of gravity of the intact vehicle, the capsule, and the remaining vehicle components are computer. The shell and plate effects are then added and the moments of inertia of the capsule and remaining vehicle are calculated. Finally, the dimensional results are determined and the program returned to the monitor.

Input Variables

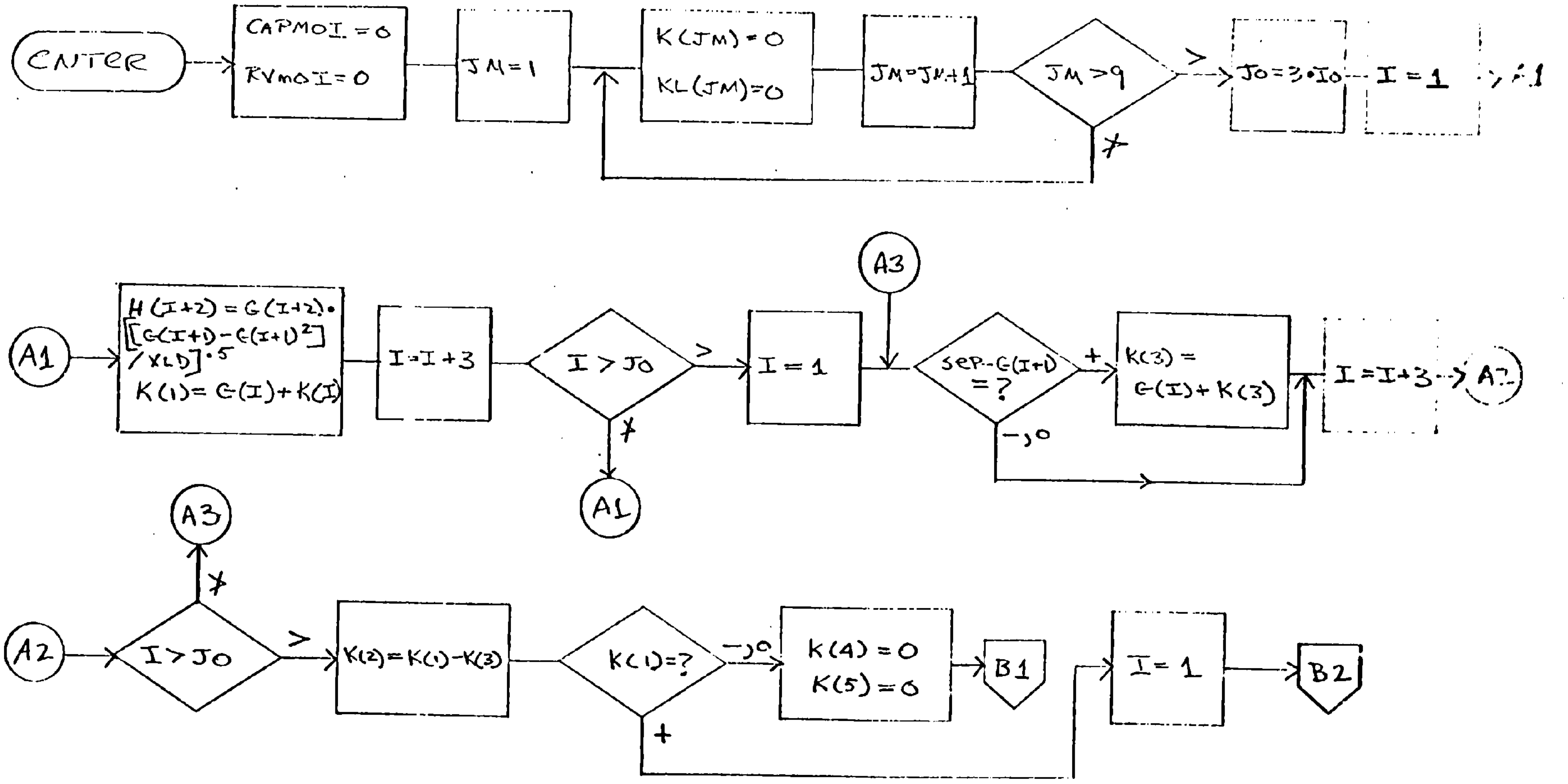
IØ	Number of weights in distribution
G	Weights and locations array (lbs., dec. frac., dec. frac.)
TCAP	Thickness of capsule shell (in.)
TRV	Thickness of remaining vehicle shell (in.)
MATCAP	Specific weight of capsule material (lbs/ft.^{3})
MATRV	Specific weight of remaining vehicle material (lbs./ft.^{3})
SEP	Separation point from nose (decimal fraction)
XLEN	Length of intact vehicle (ft.)
XLD	Length to diameter ratio
CGCAP1	Center of gravity/XLEN, capsule shell
CGRV1	Center of gravity/XLEN, remaining vehicle shell
SCAP	Surface area of capsule (ft.^{2})
SRV	Surface area of remaining vehicle (ft.^{2})

Output Variables

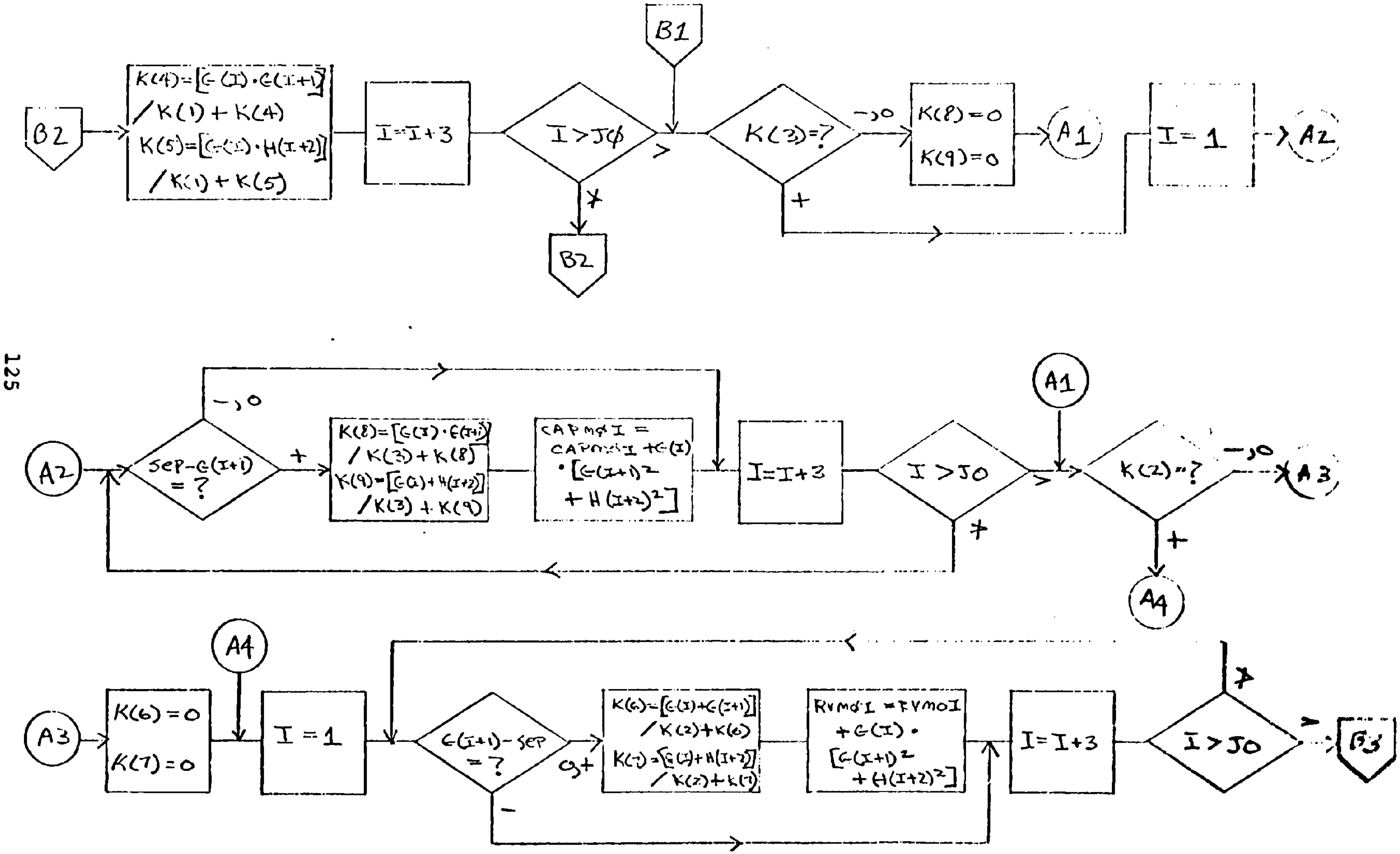
TØTWT	Weight of intact vehicle (lbs.)
RVWT	Weight of remaining vehicle (lbs.)
CAPWT	Weight of capsule (lbs.)
LCGITV	Longitudinal center of gravity of intact vehicle
VCGITV	Vertical center of gravity of intact vehicle
LCGRV	Longitudinal center of gravity of remaining vehicle
VCGRV	Vertical center of gravity of remaining vehicle
CAPMØI	Moment of inertia of capsule (slug $\cdot$ ft.^{2})
RVMØI	Moment of inertia of remaining vehicle (slug $\cdot$ ft.^{2})

IX-3.1 Flow Chart for Subroutine WGTS

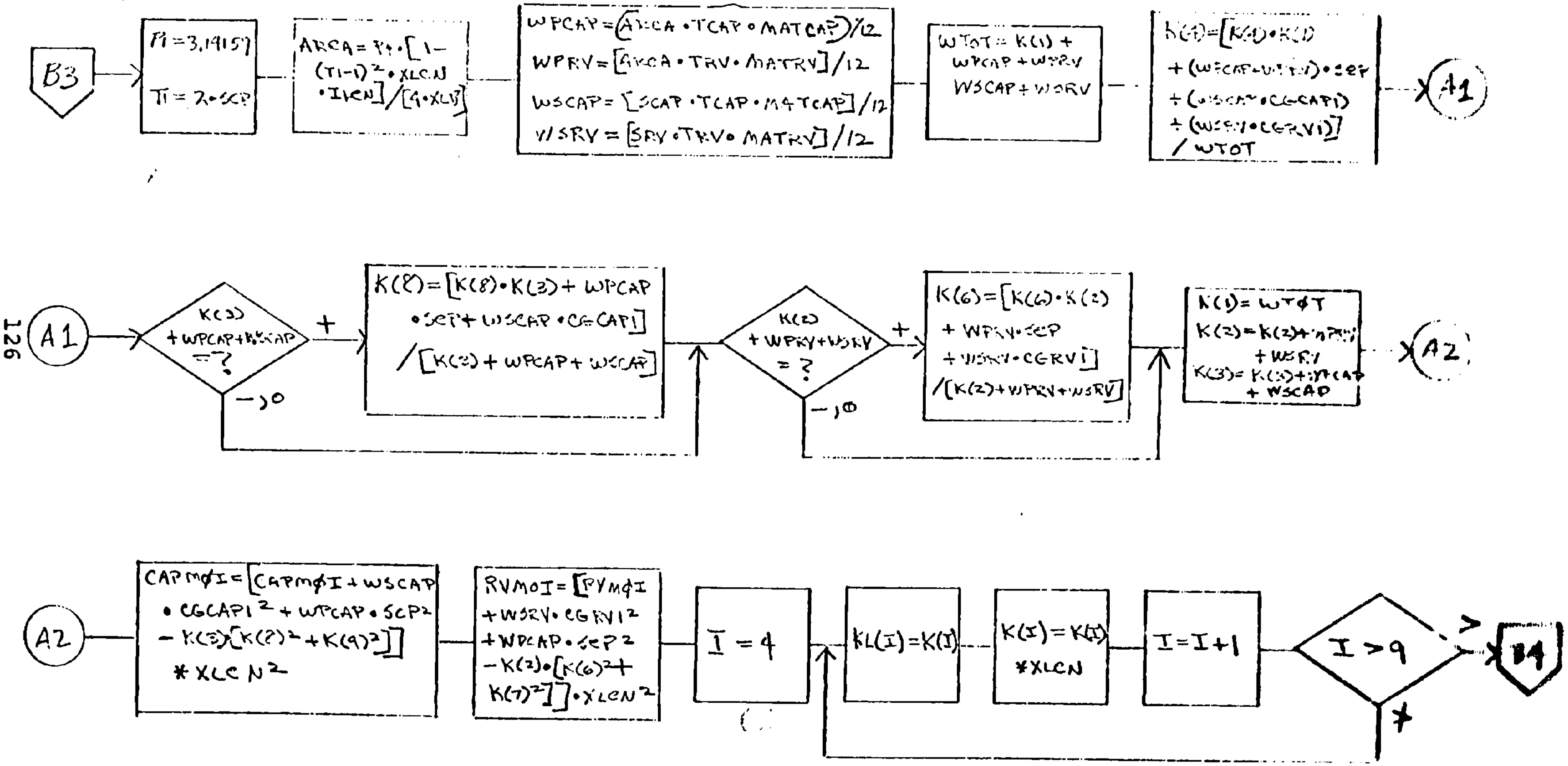
SUBROUTINE WGTS(IO,TCAP,TRV,MATCAP,MATRV,CAPMOI,RVMOI)



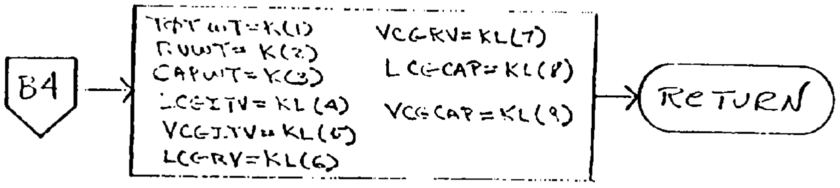
SUBROUTINE WETS (P.2)



SUBROUTINE WGTS (P.3)



SUBROUTINE WGTS (P.4)



IX-3.2 Listing of Subroutine WGTS

```
      SUBROUTINE WGTS(IO,G,ICAP,IRV,NA,CAP,NATRV,CAPMOI,RVMOI)
       IMPLICIT REAL(K,L,M)
      COMMON/INPUT/ALEN,SEP,XLD,Y3,Y4,Y5,Y6
       COMMON/INOUT/SKIP(3),CGCAPI,CGRV1,DUMK(5),SCAP,XIT(2),SR(,
     *PASS(4),TOTWT,RVWT,CAPWT,LCGITV,VCGITV,LCGRV,VCGRV,LCGCAP,
     *VCGCAP,EXTRA(707)
C
       DIMENSION K(9),G(300),KL(9),H(300)
C      CALCULATION OF TOTAL WEIGHT
C
      CAPMOI=0.0
      RVMOI=0.0
        DO 10 JM=1,9
       K(JM)=0.
 10     KL(JM)=0.
      JO=3*IO
      DO 20 I=1,JO,3
       H(I+2)=G(I+2)*(1/XLD)*(G(I+1)-G(I+1)**2)**.5
 20   K(1)= G(I)+K(1)
C
C     CALCULATION OF CAPSULE WEIGHT
C
      DO 30 I=1,JO,3
      IF (SEP-G(I+1)) 30,30,25
 25   K(3) = G(I)+K(3)
 30   CONTINUE
C
C     CALCULATION  OF REMAINING VEHICLE WEIGHT
C
      K(2) = K(1)-K(3)
C     CALCULATION OF VERT. AND LONG. CG OF INTACT VEHICLE EXCEPT
C      SHELL
       IF(K(1))100,100,110
 100   K(4)=0.
       K(5)=0.
       GO TO 120
 110   DO 40 I=1,JO,3
C
      K(4) = G(I)* G(I+1)/ K(1) +K(4)
 40   K(5)= G(I)* H(I+2)/ K(1) +K(5)
C     CALCULATION OF VERT. AND LONG. CG OF CAPSULE EXCEPT
C     SHELL
C
 120    IF(K(3))130,130,140
 130  K(8)=0.
       K(9)=0.
       GO TO 150
```

```
 140        DO 50 I=1,JO,3
        IF (SEP-G(I+1)) 50,50,45
  45        K(8) = G(I)* G(I+1)/K(3) + K(8)
        K(9) = G(I) * H(I+2)/K(3)+ K(9)
        CAPMOI=CAPMOI+G(I)*(G(I+1)**2+H(I+2)**2)
  50    CONTINUE
C
C       CALCULATION OF VERT. AND LONG. CG OF REMAINING VEHICLE
C       AFTER SHELL
 150     IF(K(2))160,160,170
 160     K(6)=0.
         K(7)=0.
         GOTO 180
C
 170        DO 60 I=1,JO,3
        IF (G(I+1) -SEP) 60,55,55
  55        K(6) = G(I)*G(I+1)/K(2) + K(6)
        K(7) = G(I)* H(I+2)/K(2) + K(7)
        RVMOI=RVMOI+G(I)*(G(I+1)**2+H(I+2)**2)
  60    CONTINUE
C
C        CALCULATION OF SHELL EFFECTS
C
 180        PI = 3.14159
        TI = 2. *SEP
        AREA = PI*((1-(TI-1)**2.)*XLEN*ILEN)/((XLD*XLD)*4.)
        WPCAP = AREA*TCAP*MATCAP/12.
        WPRV = AREA*TRV*MATRV/12.
        WSCAP = SCAP*TCAP*MATCAP/12.
        WSRV = SRV*TRV*MATRV/12.
        WTOT = K(1) + WPCAP + WPRV +WSCAP + WSRV
       K(4)=(K(4)*K(1)+(WPCAP+WPRV)*SEP+(WSCAP*CGCAP1)+(WSRV*
     1  CGRV1))/WTOT
         IF(K(3)+WPCAP+WSCAP)200,200,210
 210    K(8)=(K(8)*K(3)+WPCAP*SEP+WSCAP*CGCAP1)/(K(3)+WPCAP+WSCAP)
 200     IF(K(2)+WPRV+WSRV)220,220,230
 230    K(6)=(K(6)*K(2)+WPRV*SEP+WSRV*CGRV1)/(K(2)+WPRV+WSRV)
 220       K(1) = WTOT
        K(2) = K(2) + WPRV + WSRV
        K(3) = K(3) + WPCAP + WSCAP
C
C       ALL OF ABOVE CG'S ARE PER UNIT LENGTH BELOW WE CALC
C       DIMENSIONAL RESULTS
C
         CAPMOI=(CAPMOI+WSCAP*CGCAP1*CGCAP1+WPCAP*SEP*SEP-K(3)
     **(K(8)**2+K(9)**2))*XLEN*XLEN
```

```
      RVMOI=(RVMOI+WSRV*CGRV1*CGRV1+WPCAP*SEP*SEP-K(2)
     **(K(6)**2+K(7)**2))*XLEN*XLEN

      DO 70 I=4,9
      KL(I) = K(I)
70    K(I) = K(I) * XLEN
      TOTWT=K(1)
      RVWT=K(2)
      CAPWT=K(3)
      LCGITV=KL(4)
      VCGITV=KL(5)
      LCGRV=KL(6)
      VCGRV=KL(7)
      LCGCAP=KL(8)
      VCGCAP=KL(9)
      RETURN
      END
```

IX-4. Subroutine CAS

The CAS subroutine is called by the command DAMAGE) and simulates flooding the specific floodable volume(s) with sea water. These new weights are then added to the present distribution along with their respective logitudinal and vertical centers.

There are no error messages.

Input Variables

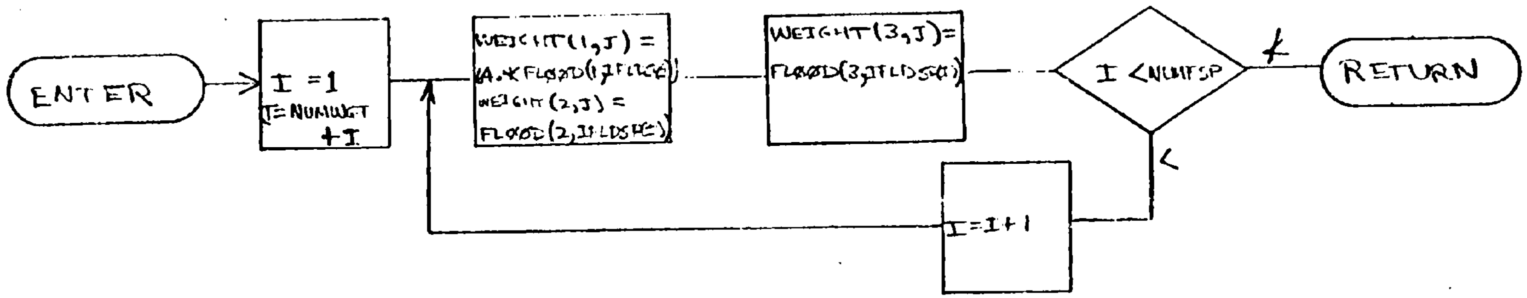
WEIGHT	Weight array (lbs., dec. frac., dec. frac.)
NUMWGT	Number of weights (triples) in array, WEIGHT
FLØØD	Floodable volume array ($ft.^3$, dec. frac., dec. frac.)
NUMFLD	Number of volumes (triples) in array FLØØD
IFLDSP	Array containing identification of spaces to be flooded
NUMFSP	Number of spaces to be flooded

Output Variables

WEIGHT	Extended weight array (lbs., dec. frac., dec. frac.)

IX-4.1 Flow Chart for Subroutine CAS

SUBROUTINE CAS



IX-4.2 Listing of Subroutine CAS

```
      SUBROUTINE CAS(WEIGHT,NUMWGT,FLOOD,NUMFLD,IFLDSP,NUMFSP)
      DIMENSION WEIGHT(3,100),FLOOD(3,100),IFLDSP(10)
      DO 100 I=1,NUMFSP
      J=NUMWGT+I
      WEIGHT(1,J)=64.*FLOOD(1,IFLDSP(I))
      WEIGHT(2,J)=FLOOD(2,IFLDSP(I))
100   WEIGHT(3,J)=FLOOD(3,IFLDSP(I))
      RETURN
      END
```

IX-5. Subroutine HST1

Inputs for the hydrostatics program include the weight, volume, longitudinal and vertical centers of gravity, longitudinal center of buoyancy, and initial trim of the intact vehicle. If the vehicle is damaged (IDMSW=0), calculations are made in the first half of the program primarily between statements 2000 and 2080. If the vehicle is damaged (IDMSW=1), the calculations are performed between 2110 and 2200.

The subroutine is called in response to the monitor command TRIM) to evaluate the hydrostatic characteristics of the undamaged or damaged intact vehicle designed by the user. It outputs no data for use by any other subroutine.

There are no error messages as such; however, the messages printed out are meant as an intuitive design guide for the user.

Input Variables

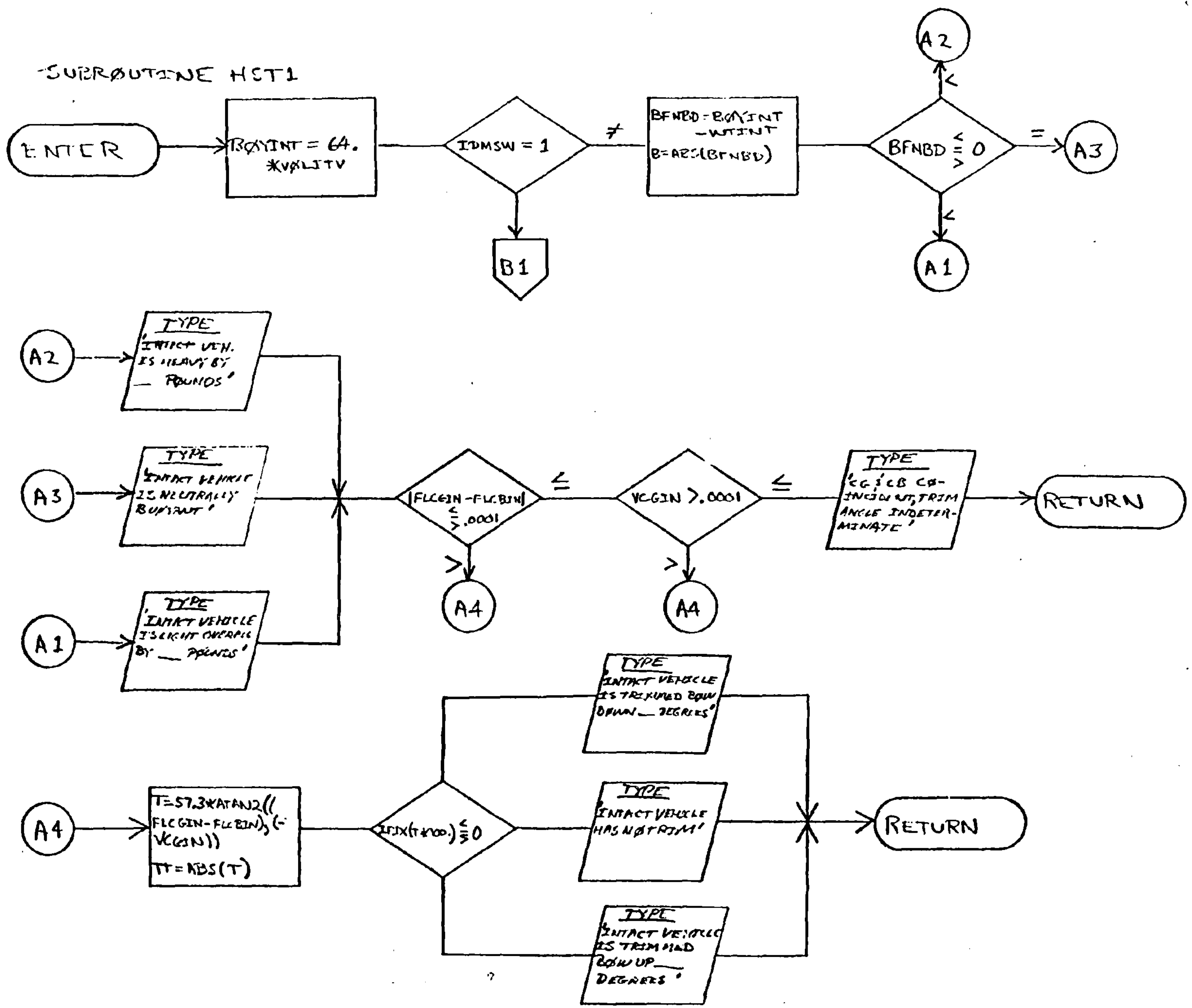
IDMSW	Damage switch
FLCBIN	Longitudinal center of buoyancy/L, intact vehicle
VØLITV	Volume of intact vehicle ($ft.^3$)
WTINT	Weight of intact vehicle (lbs.)
FLGGIN	Longitudinal center of gravity/L, intact vehicle
VCGIN	Vertical center of gravity/L

Output Variables

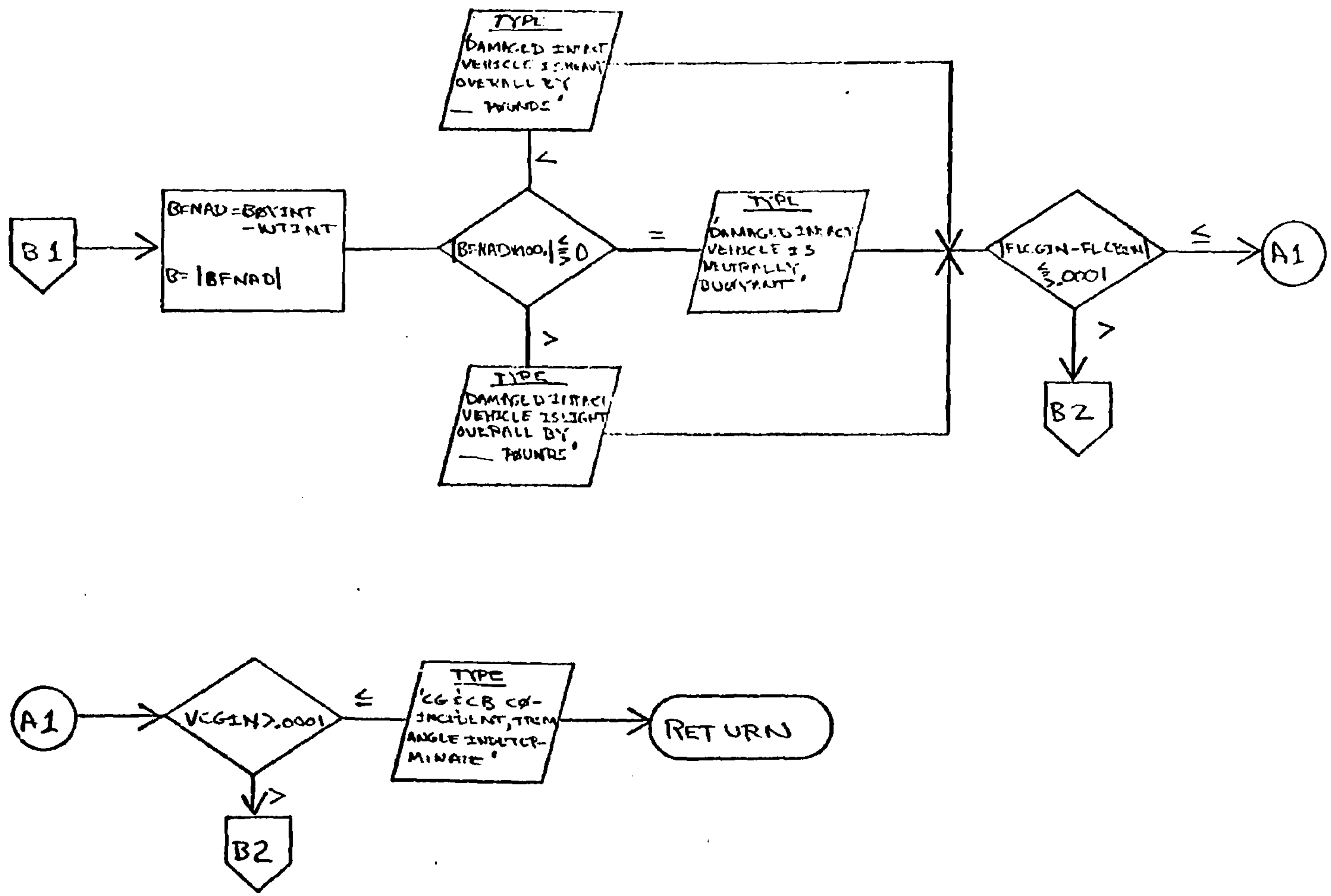
BFNBD	Net buoyant force of intact vehicle before damage (lbs.)
TT	Bow down trim of intact vehicle (deg.)
T	Bow up trim of intact vehicle (deg.)
BFNAD	Net buoyant force of intact vehicle after damage (lbs.)
TAD	Bow down trim of damaged intact vehicle (deg.)
TADIN	Bow up trim of damaged intact vehicle (deg.)

IX-5.1 Flow Chart for Subroutine HST1

SUBRØUTINE HST1

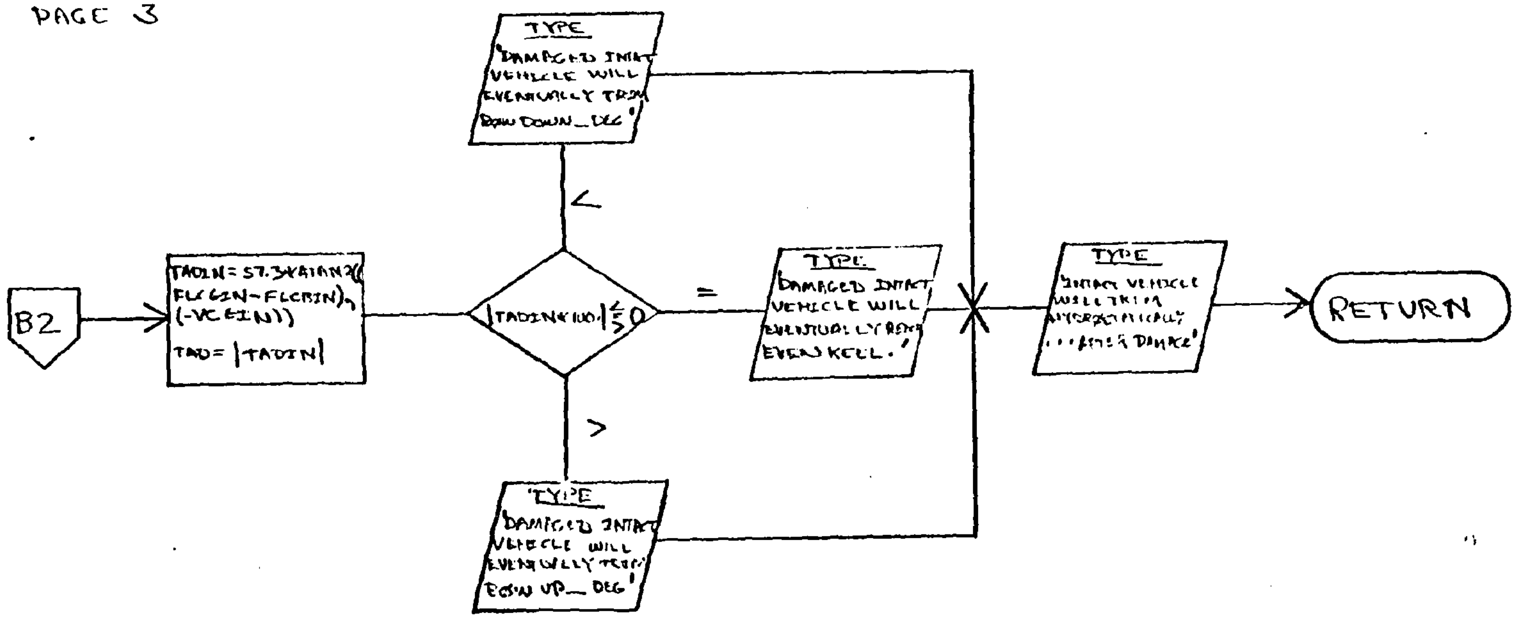


SUBROUTINE KST1
PAGE 2



SUBROUTINE HST1

PAGE 3



IX-5.2 Listing of Subroutine HST1

```
      SUBROUTINE HST1(IDMSW)
      COMMON/INPUT/FLOA,PASS(2),R
      COMMON/INOUT/FLCBIN,SKIP(4),VOLITV,STUFF(12),WTINT,
     * FOOL(2),FLCGIN,VCGIN,EXTRA(711)
C********************************
C
C  CALL TO OVERFLOW(-I) SUPPRESSES DIVIDE CHECK ERROR MESSAGES.
C
C********************************
      CALL OVERFL(O)
      BOYINT=64.*VOLITV
      GO TO (2110),IDMSW
      BFNBD=BOYINT-WTINT
       B=ABS(BFNBD)
      IF(BFNBD)2000,2010,2020
 2000 TYPE 2990,B
 2990 FORMAT(36H INTACT VEHICLE IS HEAVY OVERALL BY ,F10.0,
     1 7H POUNDS)
      GO TO 2030
 2010 TYPE 2980
 2980 FORMAT(36H INTACT VEHICLE IS NEUTRALLY BUOYANT)
      GO TO 2030
 2020 TYPE 2970,BFNBD
 2970 FORMAT(36H INTACT VEHICLE IS LIGHT OVERALL BY ,F10.0,
     1 7H POUNDS)
 2030 IF(ABS(FLCGIN-FLCBIN)-.0001)3000,3000,2031
 3000  IF(VCGIN-.0001)3010,3010,2031
 3010 TYPE 3020
 3020 FORMAT(' CG AND CB COINCIDENT, TRIM ANGLE INDETERMINATE'/)
      RETURN
2031       T=57.2958*ATAN2((FLCGIN-FLCBIN),(-VCGIN))
      TT=ABS(T)
 2040 IF(IFIX(T*100.))2050,2060,2070
 2050 TYPE 2960,TT
 2960 FORMAT(36H INTACT VEHICLE IS TRIMMED BOW DOWN ,F7.2,
     1 8H DEGREES)
      RETURN
 2060 TYPE 2950
 2950  FORMAT(27H INTACT VEHICLE HAS NO TRIM)
      RETURN
 2070 TYPE 2940,T
 2940 FORMAT(34H INTACT VEHICLE IS TRIMMED BOW UP ,F7.2,
     1 'DEGREES')
 2080 RETURN
C      THE ABOVE ESTABLISH THE BEFORE DAMAGE CONDITION OF THE
C      INTACT VEHICLE
```

```
2110   BFNAD=BOYINT-WTINT
       B=ABS(BFNAD)
       IF(IFIX(BFNAD*100.))2120,2130,2140
2120  TYPE 2920,B
2920   FORMAT(44H DAMAGED INTACT VEHICLE IS HEAVY OVERALL BY ,
     1 F10.0, 7H POUNDS)
       GO TO 2150
2130   TYPE 2910
2910   FORMAT(44H DAMAGED INTACT VEHICLE IS NEUTRALLY BUOYANT)
       GO TO 2150
2140   TYPE 2900,BFNAD
2900   FORMAT(44H DAMAGED INTACT VEHICLE IS LIGHT OVERALL BY ,
     1 F10.0, 7H POUNDS)
2150   IF(ABS(FLCGIN-FLCBIN)-.0001)4000,4000,2160
4000   IF(VCGIN-.0001)4010,4010,2160
4010   TYPE 4020
4020   FORMAT(' CG AND CB COINCIDENT, TRIM ANGLE INDETERMINATE'/)
       RETURN
2160    TADIN=57.2958*ATAN2((FLCGIN-FLCBIN),(-VCGIN))
       TAD=ABS(TADIN)
       IF(IFIX(TADIN*100.))2170,2180,2190
2170   TYPE 2890,TAD
2890   FORMAT(54H DAMAGED INTACT VEHICLE WILL EVENTUALLY TRIM
     1 BOW DOWN , F7.2,8H DEGREES)
        GO TO 2200
2180   TYPE 2880
2880   FORMAT(55H DAMAGED INTACT VEHICLE WILL EVENTUALLY REACH
     1 EVEN KEEL)
       GO TO 2200
2190   TYPE 2870,TADIN
2870   FORMAT(52H DAMAGED INTACT VEHICLE WILL EVENTUALLY TRIM
     1 BOW UP , F7.2,8H DEGREES)
2200       TYPE 2860, T,TADIN
2860   FORMAT(47H INTACT VEHICLE WILL TRIM HYDROSTATICALLY FROM
     1 ,F7.2,/,26H DEGREES BEFORE DAMAGE TO ,F7.2,14H DEGREES AFTER
     2 ' DAMAGE')
2210   RETURN
        END
```

IX-6. Trajectory Solution (MØT)

This subroutine calculates the trajectory of either the escape capsule or the remaining vehicle depending on whether it was called by typing the command ASCEND) or the command HULL).

The subroutine allows the user to specify the upper limit of the time interval in seconds. This value is non-dimensionalized internally and calculations proceed for evaluating the distances, angles and velocities of the vehicle from the point of separation subject to the initial conditions of speed, trim and physical characteristics of the vehicle input or calculated in the monitor or in preceding subroutines.

Error Messages

1) "THE VEHICLE IS DYNAMICALLY UNSTABLE...INTEGRATION HAS BEEN TERMINATED. RESULTS MAY BE PRINTED OR GRAPHED." - An instability has been detected and time histories have been terminated. The histories may be printed or plotted.

Corrective Action: Change input parameters to try to obtain stable solution if desired.

Input Variables

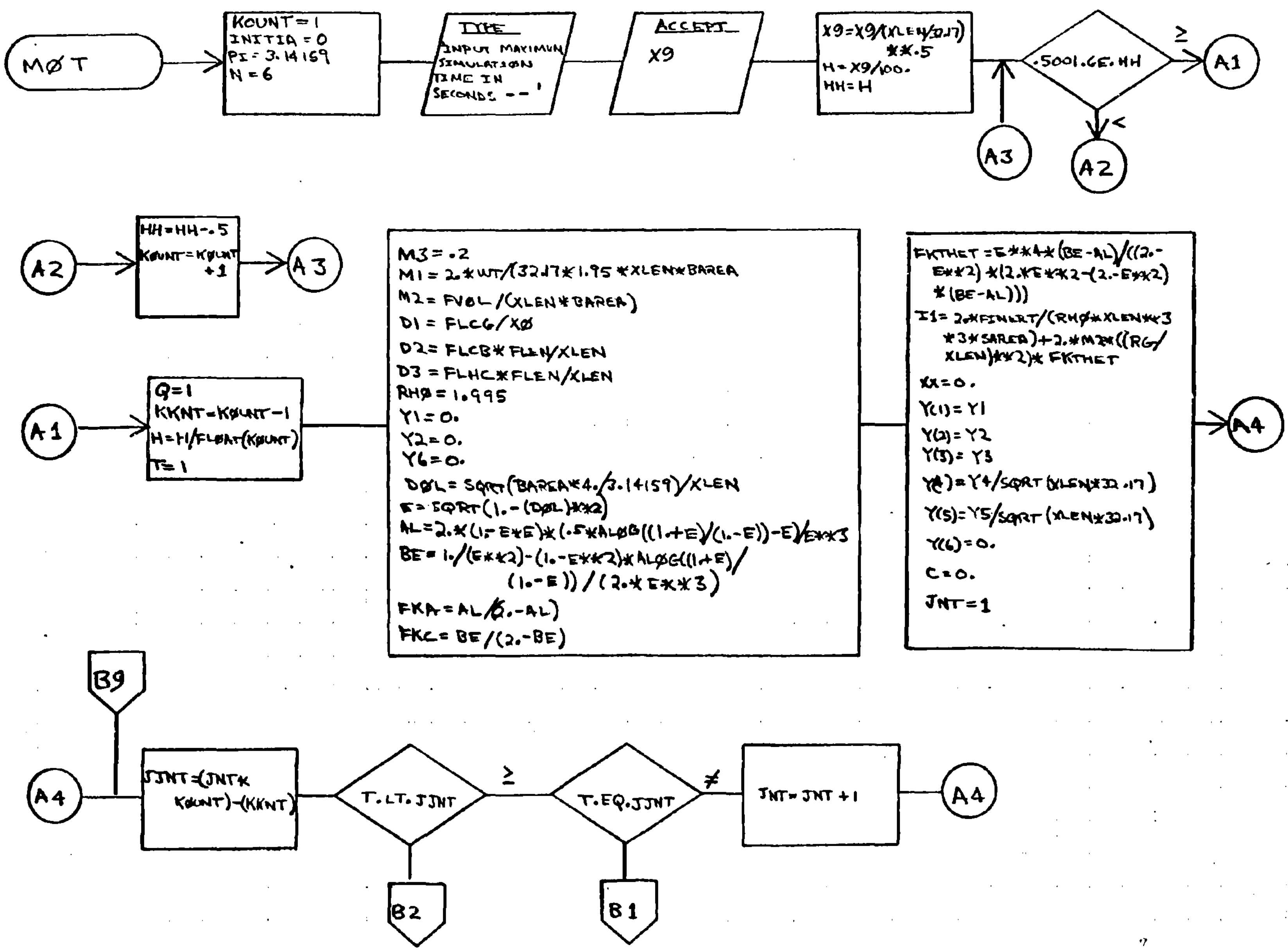
XLEN	Length of vehicle (ft.)
WT	Weight of vehicle (lbs.)
FVØL	Volume of vehicle (ft^3.)
FINERT	Moment of inertia of vehicle ($slug \cdot ft^2$.)
RAD	Radius of gyration of vehicle (ft.)
SAREA	Surface area of vehicle (ft^2.)
FLCG	Longitudinal center of gravity of vehicle/XLEN
FVCG	Vertical center of gravity of vehicle/XLEN
FLCB	Longitudinal center of buoyancy/XLEN
FLHC	Center of pressure of projected vehicle area/XLEN
BAREA	Basic area of vehicle (ft^2.)
Y3	Initial trim angle (deg.)
Y4	Initial vertical velocity (ft./sec.)
Y5	Initial horizontal velocity (ft./sec.)

Output Variables

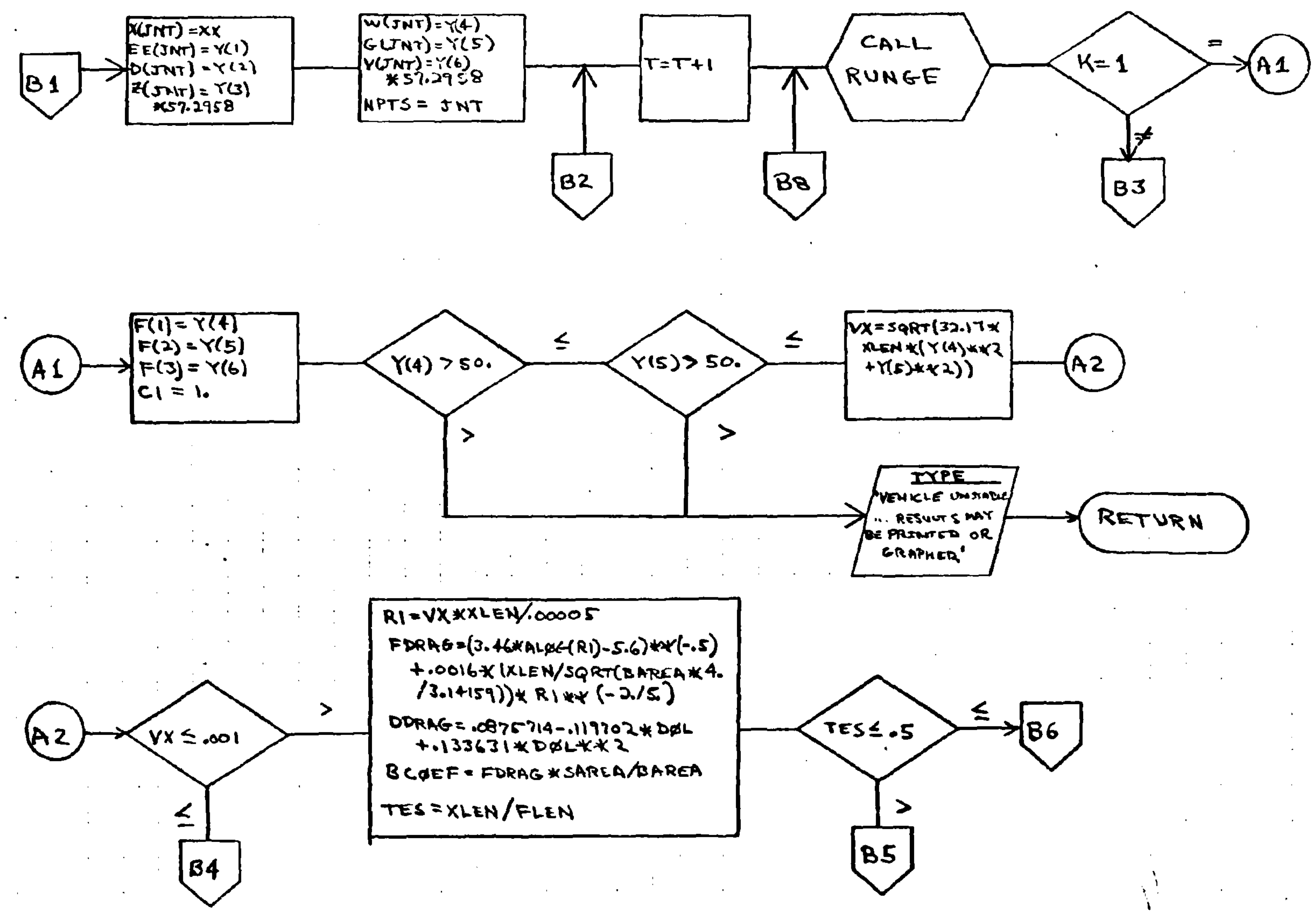
X	Time array, non-dimensional
Z	Attitude array (deg.)
D	Vertical distance array, non-dimensional
E	Horizontal distance array, non-dimensional
V	Angular velocity array (deg./non-dimen.time)
G	Vertical velocity array, non-dimensional
W	Horizontal velocity array, non-dimensional
NPTS	Number of points in each array

IX-6.1 Flow Chart for Trajectory Solution (MØT)

Pg 1

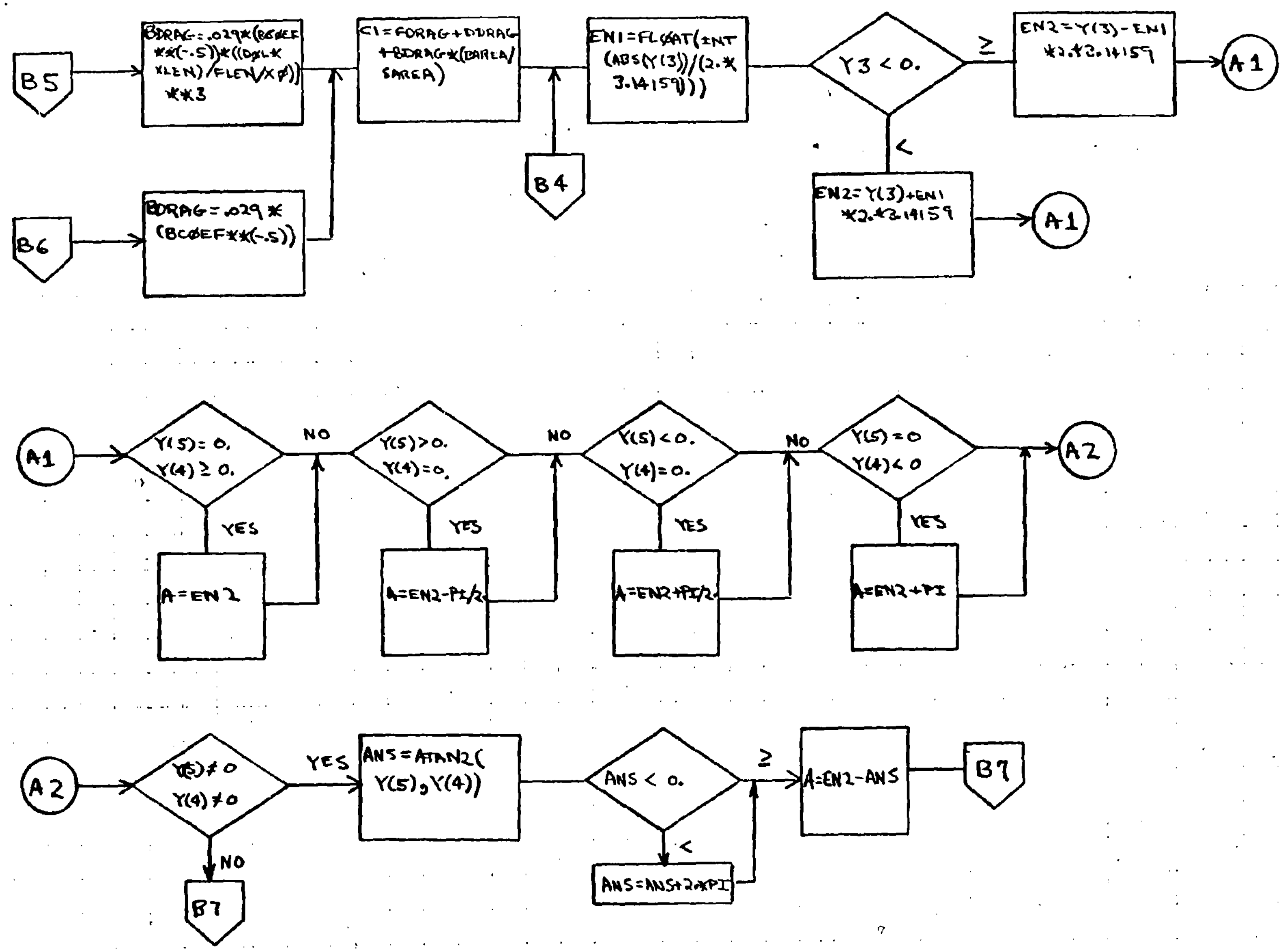

MØT
KOUNT=1
INITIA=0
PI=3.14159
N=6
TYPE
INPUT MAXIMUM SIMULATION TIME IN SECONDS --'
ACCEPT
X9
X9=X9/(XLEN/32.17)**.5
H=X9/100.
HH=H
.5001.GE.HH
≥
<
A1
A2
A3
HH=HH-.5
KOUNT=KOUNT+1
Q=1
KKNT=KOUNT-1
H=H/FLOAT(KOUNT)
T=1
M3=.2
M1=2.*WT/(32.17*1.95*XLEN*BAREA
M2=FVOL/(XLEN*BAREA)
D1=FLCG/XØ
D2=FLCB*FLEN/XLEN
D3=FLHC*FLEN/XLEN
RHØ=1.995
Y1=0.
Y2=0.
Y6=0.
DØL=SQRT(BAREA*4./3.14159)/XLEN
E=SQRT(1.-(DØL)**2)
AL=2.*(1.-E*E)*(.5*ALØG((1.+E)/(1.-E))-E)/E**3
BE=1./(E**2)-(1.-E**2)*ALØG((1+E)/(1.-E))/(2.*E**3)
FKA=AL/(2.-AL)
FKC=BE/(2.-BE)
FKTHET=E**4*(BE-AL)/((2.-E**2)*(2.*E**2-(2.-E**2)*(BE-AL)))
I1=2.*FINERT/(RHØ*XLEN**3*3*SAREA)+2.*M2*((RG/XLEN)**2)*FKTHET
XX=0.
Y(1)=Y1
Y(2)=Y2
Y(3)=Y3
Y(4)=Y4/SQRT(XLEN*32.17)
Y(5)=Y5/SQRT(XLEN*32.17)
Y(6)=0.
C=0.
JNT=1
A4
B9
JJNT=(JNT*KOUNT)-(KKNT)
T.LT.JJNT
≥
T.EQ.JJNT
≠
JNT=JNT+1
B2
B1

pg 2

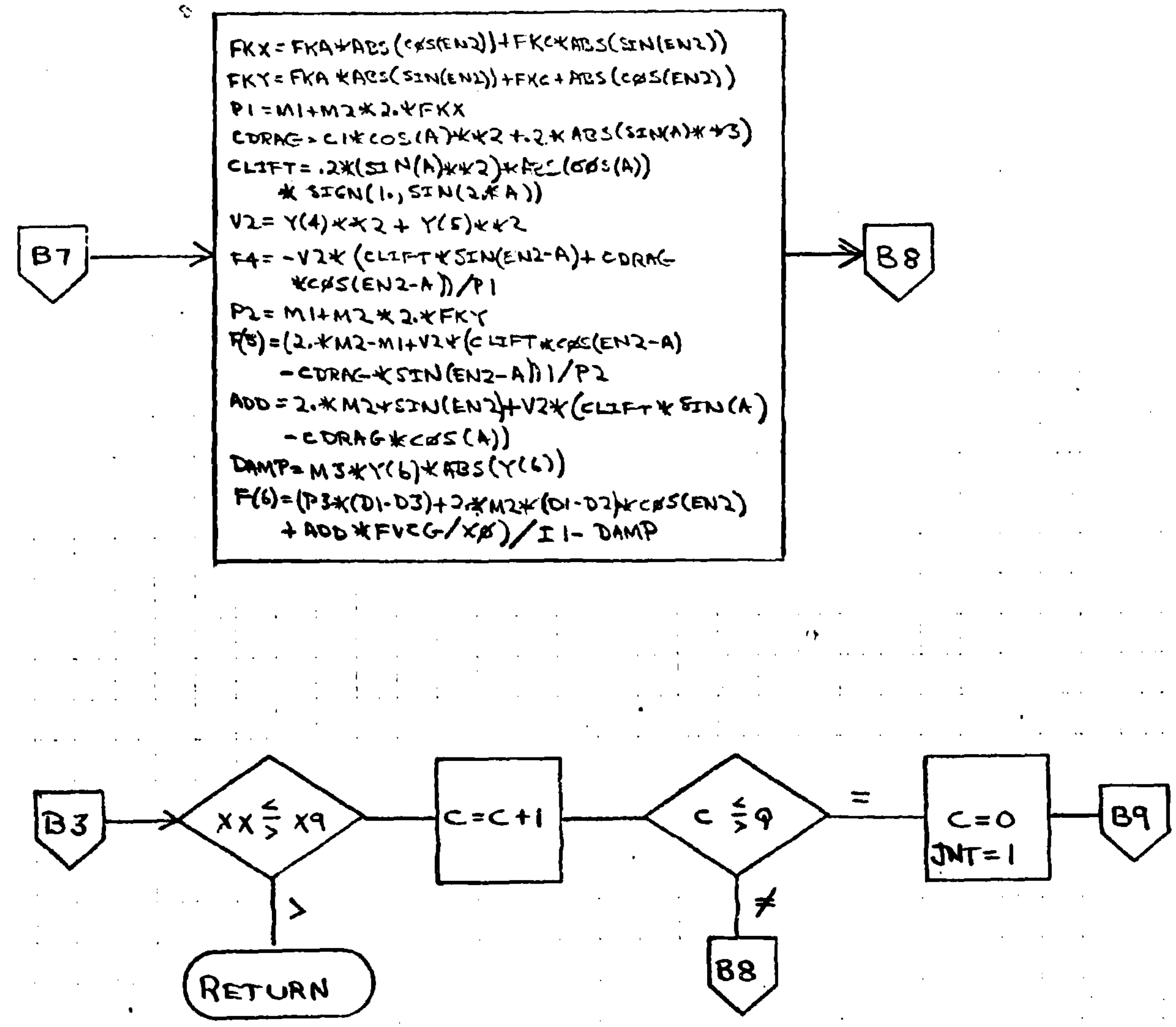


B1
X(JNT) = XX
EE(JNT) = Y(1)
D(JNT) = Y(2)
Z(JNT) = Y(3) *57.2958
W(JNT) = Y(4)
G(JNT) = Y(5)
V(JNT) = Y(6) *57.2958
NPTS = JNT
B2
T=T+1
B8
CALL RUNGE
K=1
=
≠
A1
B3
A1
F(1) = Y(4)
F(2) = Y(5)
F(3) = Y(6)
C1 = 1.
Y(4) > 50.
≤
>
Y(5) > 50.
≤
>
VX=SQRT(32.17*XLEN*(Y(4)**2+Y(5)**2))
A2
TYPE
'VEHICLE UNSTABLE ... RESULTS MAY BE PRINTED OR GRAPHED'
RETURN
A2
VX ≤ .001
≤
B4
>
R1=VX*XLEN/.00005
FDRAG=(3.46*ALOG(R1)-5.6)**(-.5) +.0016*(XLEN/SQRT(BAREA*4./3.14159))*R1**(-2./5.)
DDRAG=.0875714-.119702*DOL +.133631*DOL**2
BCOEF=FDRAG*SAREA/BAREA
TES=XLEN/FLEN
TES ≤ .5
≤
B6
>
B5

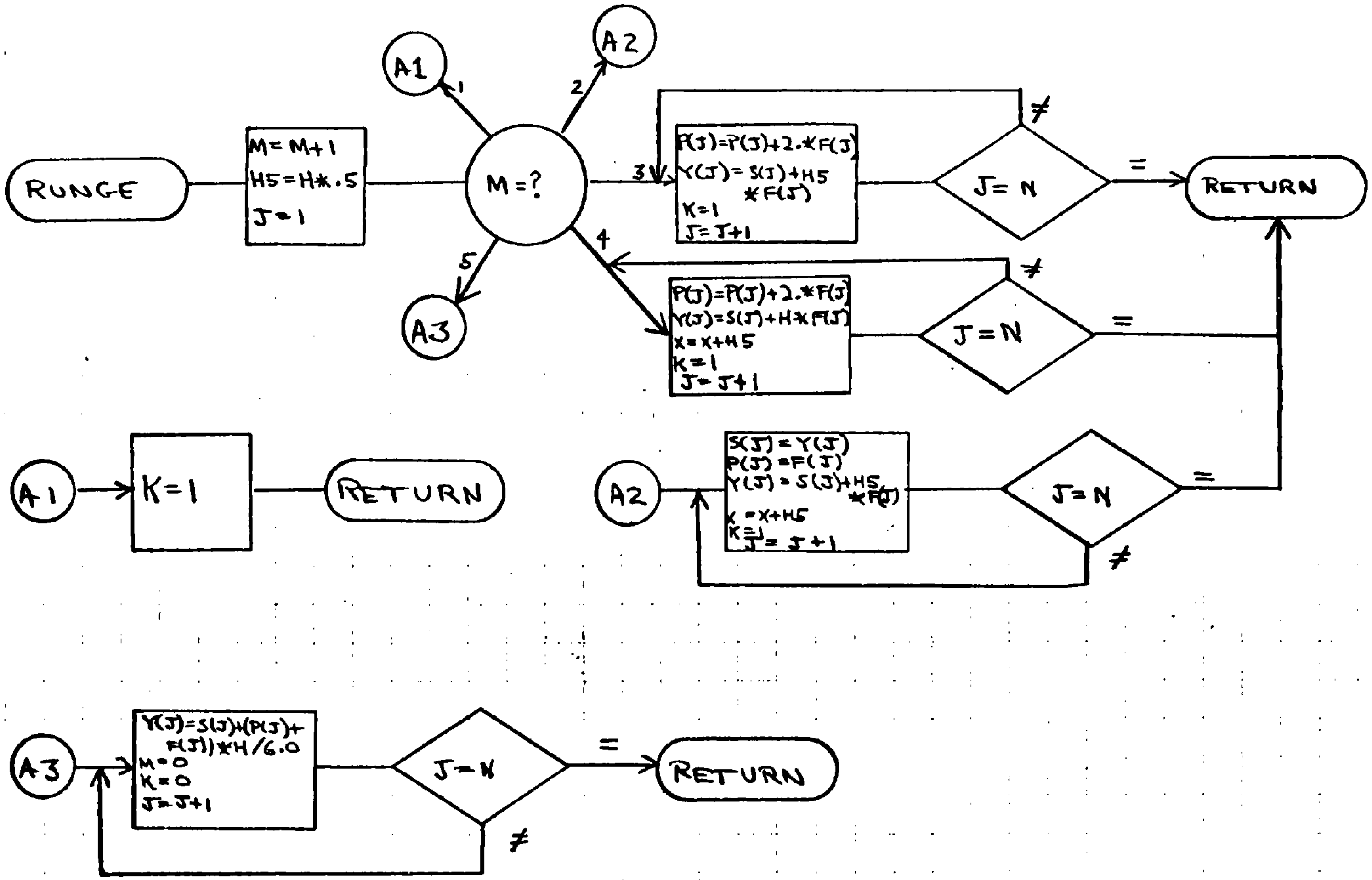
pg 3


B5
BDRAG=.029*(BCØEF**(-.5))*((DØL*XLEN)/FLEN/XØ))**3
B6
BDRAG=.029*(BCØEF**(-.5))
C1=FDRAG+DDRAG+BDRAG*(DAREA/SAREA)
B4
EN1=FLØAT(INT(ABS(Y(3))/(2.*3.14159)))
Y3 < 0.
≥
EN2=Y(3)-EN1*2.*3.14159
A1
<
EN2=Y(3)+EN1*2.*3.14159
A1
A1
Y(5)=0. Y(4) ≥ 0.
YES
A=EN2
NO
Y(5)>0. Y(4)=0.
YES
A=EN2-PI/2
NO
Y(5)<0. Y(4)=0.
YES
A=EN2+PI/2
NO
Y(5)=0 Y(4)<0
YES
A=EN2+PI
A2
A2
Y(5)≠0 Y(4)≠0
NO
B7
YES
ANS=ATAN2(Y(5),Y(4))
ANS < 0.
<
ANS=ANS+2.*PI
≥
A=EN2-ANS
B7

PJ4


B7
FKX = FKA*ABS(COS(EN2))+FKC*ABS(SIN(EN2))
FKY = FKA*ABS(SIN(EN2))+FKC*ABS(COS(EN2))
P1 = M1+M2*2.*FKX
CDRAG = C1*COS(A)**2+.2*ABS(SIN(A)**3)
CLIFT = .2*(SIN(A)**2)*ABS(COS(A))
* SIGN(1.,SIN(2.*A))
V2 = Y(4)**2 + Y(5)**2
F4 = -V2*(CLIFT*SIN(EN2-A)+CDRAG
*COS(EN2-A))/P1
P2 = M1+M2*2.*FKY
F(5) = (2.*M2-M1+V2*(CLIFT*COS(EN2-A)
-CDRAG*SIN(EN2-A)))/P2
ADD = 2.*M2*SIN(EN2)+V2*(CLIFT*SIN(A)
-CDRAG*COS(A))
DAMP = M3*Y(6)*ABS(Y(6))
F(6) = (P3*(D1-D3)+2.*M2*(D1-D2)*COS(EN2)
+ADD*FVEG/XØ)/I1-DAMP
B8
B3
XX ≤ ≥ X9
>
RETURN
C=C+1
C ≤ ≥ 9
=
≠
B8
C=0
JNT=1
B9

pg 5



RUNGE
M=M+1
H5=H*.5
J=1
M=?
A1
A2
A3
1
2
3
4
5
P(J)=P(J)+2.*F(J)
Y(J)=S(J)+H5
*F(J)
K=1
J=J+1
J=N
=
≠
RETURN
P(J)=P(J)+2.*F(J)
Y(J)=S(J)+H*F(J)
X=X+H5
K=1
J=J+1
J=N
=
≠
A1
K=1
RETURN
A2
S(J)=Y(J)
P(J)=F(J)
Y(J)=S(J)+H5
*F(J)
X=X+H5
K=1
J=J+1
J=N
=
≠
A3
Y(J)=S(J)+(P(J)+
F(J))*H/6.0
M=0
K=0
J=J+1
J=N
=
≠
RETURN

IX-6.2 Listing of Trajectory Solutions (MØT)

```
      SUBROUTINE MOT(XLEN,WT,FVOL,FINERT,RG,SAREA,FLCG,FVCG,FLCB,FLHC,
     *NPTS)
      REAL M1,M2,M3,I1
      COMMON/INPUT/FLEN,XO,FLDRAT,Y3,Y4,Y5,Y6
      COMMON/RUNG/INITIA
      COMMON/INOUT/EXTRA(15),BAREA,SKIP(11),X(101),Z(101),
     *D(101),EE(101),V(101),G(101),W(101)
      DIMENSION F(6),Y(6)
      INTEGER T,C,Q,RUNGE
      KOUNT=1
      INITIA=0
      PI=3.14159
30    N=6
      TYPE 3000
3000  FORMAT(' INPUT MAXIMUM SIMULATION TIME IN SECONDS -- '$)
      ACCEPT3001,X9
3001  FORMAT(F)
      X9=X9/(XLEN/32.17)**.5
      H=X9/100.
      HH=H
10    IF(.5001.GE.HH)GO TO 60
      HH=HH-.5
      KOUNT=KOUNT+1
      GO TO 10
60    Q=1
      KKNT=KOUNT-1
      H=H/FLOAT(KOUNT)
      T=1
       M3=0.2
      M1=2.*WT/(32.17*1.95*XLEN*BAREA)
      M2=FVOL/(XLEN*BAREA)
      D1=FLCG/XO
      D2=FLCB*FLEN/XLEN
      D3=FLHC*FLEN/XLEN
      RHO=1.995
      Y1=0.
      Y2=0.
      Y6=0.
      DOL=SQRT(BAREA*4./3.14159)/XLEN
      E=SQRT(1.-(DOL)**2)
      AL=2.*(1.-E*E)*(.5*ALOG((1.+E)/(1.-E))-E)/E**3
      BE=1./(E**2)-(1.-E**2)*ALOG((1.+E)/(1.-E))/(2.*E**3)
      FKA=AL/(2.-AL+
     -FKC=BE/(2.-BE)
      FKTHET=E**4*(BE-AL)/((2.-E**2)*(2.*A-A
E**2-(2.-E**2)*(BE-AL)))
      I1=2.*FINERT/(RHO*XLEN**3*SAREA)+2.*M2*((RG/XLEN)**2)*FKTHET
      XX=0.
```

```
 360   Y(1)=Y1
 370   Y(2)=Y2
 380   Y(3)=Y3
 390   Y(4)=Y4/SQRT(32.17*XLEN)
 400   Y(5)=Y5/SQRT(32.17*XLEN)
 410   Y(6)=0.
 430   C=0
 510      DO 8 JNT=1,101
        JJNT=(JNT*KOUNT)-(KKNT)
        IF(T.LT.JJNT) GO TO 555
        IF(T.EQ.JJNT)GO TO 520
 8      CONTINUE
 520    X(JNT)=XX
       EE(JNT)=Y(1)
 525   D(JNT)=Y(2)
 530   Z(JNT)

=Y(3)*57.2958
 535   W(JNT)=Y(4)
 540   G(JNT)=Y(5)
 545   V(JNT)=Y(6)*57.2958
 550    NPTS=JNT-1
 555       T=T+1
 560    CALL RUNGE(6,Y,F,XX,H,K)
 580   IF(K-1)750,600,750
 600   F(1)=Y(4)
 610   F(2)=Y(5)
 620   F(3)=Y(6)
        C1=1.
        IF(Y(4).GT.50.)GOTO 2000
        IF(Y(5).GT.50.)GO TO 2000
 621     VX=SQRT(32.17*XLEN*(Y(4)**2+Y(5)**2))
        IF(VX-.001)633,633,622
 622    R1=VX*XLEN/1.0E-5
 624     FDRAG=(3.46*ALOG(R1)-5.6)**(-.5)
    *+0.0016*(XLEN/SQRT(BAREA*4./3.14159))*R1**(-2./5.)
 625    DDRAG=.0875714-0.119702*DOL+0.133631*DOL**2
 626    BCOEF=FDRAG*SAREA/BAREA
 627     TES=XLEN/FLEN
 628    IF (TES.LE.0.5) GOTO 631
 629    BDRAG=0.029*(BCOEF**(-.5))*((DOL*XLEN)/(FLEN/XO))**3
 630    GOTO 632
 631    BDRAG=0.029*(BCOEF**(-.5))
632     C1=FDRAG+DDRAG+BDRAG*(BAREA/SAREA)
 633     CONTINUE
          EN1=FLOAT(INT(ABS(Y(3))/(2.*3.14159)))
         IF(Y(3).LT.0.) GO TO 650
         EN2=Y(3)-EN1*2.*3.14159
         GO TO 660
 650     EN2=Y(3)+EN1*2.*3.14159
 660    IF(Y(5).EQ.0.0.AND.Y(4).GE.0.0) A=EN2
        IF(Y(5).GT.0.0.AND.Y(4).EQ.0.0) A=EN2-PI/2.
```

```
      IF(Y(5).LT.0.0.AND.Y(4).EQ.0.0) A=EN2+PI/2.
      IF(Y(5).EQ.0.0.AND.Y(4).LT.0.0) A=EN2+PI
      IF(Y(5).NE.0.0.AND.Y(4).NE.0.0) GO TO 666
      GO TO 670
666   ANS=ATAN2(Y(5),Y(4))
      IF(ANS.LT.0.0)  ANS=ANS+2.*PI
      A=EN2-ANS
670   FKX=FKA*ABS(COS(EN2))+FKC*ABS(SIN(EN2))
      FKY=FKA*ABS(SIN(EN2))+FKC*ABS(COS(EN2))
         P1=M1+M2*2.*FKX
      CDRAG=C1*COS(A)**2+.2*ABS(SIN(A)**3)
      CLIFT=.2*(SIN(A)**2)*ABS(COS(A))*SIGN(1.,SIN(2..A))
      V2=Y(4) **2+Y(5)**2
      F(4)=-V2*(CLIFT*SIN(EN2-A)+CDRAG*COS(EN2-A))/P1
      P2=M1+M2*2.*FKY
      F(5)=(2.*M2-M1+V2*(CLIFT*COS(EN2-A)-CDRAG*SIN(EN2-A)))/P2
      P3=V2*(CDRAG*SIN(A)+CLIFT*COS(A))


      ADD=2.*M2*SIN(EN2)+V2*(CLIFT*SIN(A)-CDRAG*COS(A))
      DAMP=M3*Y(6)*ABS(Y(6))
      F(6)=(P3*(D1-D3)+2.*M2*(D1-D2)*COS(EN2)+ADD*FVCG/X0)/I1-DAMP
740   GO TO 560
750   IF(XX-X9)780,780,770
770   GO TO 815
780   C=C+1
790   IF(C-Q)560,800,560
800   C=0
810   GO TO 510
2000   TYPE 2001
2001   FORMAT(' THE VEHICLE IS DYNAMICALLY UNSTABLE...INTEGRATION HAS',
    */,' BEEN TERMINATED.  RESULTS MAY BE PRINTED OR GRAPHED.',//)
815    RETURN
816   END
```

```
820    SUBROUTINE RUNGE(N,Y,F,X,H,K)
822   DIMENSION P(50),S(50),Y(6),F(6)
      COMMON/RUNG/M
828  M=M+1
      H5=H*.5
830   GO TO (880,900,980,1040,1110),M
880   K=1
890  RETURN
900  DO 930 J=1,N
910  S(J)=Y(J)
920  P(J)=F(J)
930  Y(J)=S(J)+H5*F(J)
950  X=X+H5
960    K=1
970  RETURN
980  DO 1000 J=1,N
990  P(J)=P(J)+2.*F(J)
1000 Y(J)=S(J)+H5*F(J)
1020   K=1
1030 RETURN
1040 DO 1060 J=1,N
1050 P(J)=P(J)+2.*F(J)
1060 Y(J)=S(J)+H*F(J)
1080 X=X+H5
1090   K=1
1100 RETURN
1110 DO 1120 J=1,N
1120 Y(J)=S(J)+(P(J)+F(J))*H/6.0
1140 M=0
1150  K=0
1160 RETURN
1170 END
```

IX-7. Subroutine PLØT

This package is called in response to the monitor command GRAPH). It furnishes the user with annotated plot of the vehicle's non-dimensional vertical and horizontal distances from the point of separation (time = 0) and of its attitude in degrees versus the independent variable, non-dimensionalized time. The vehicle may be either the capsule or the remaining vehicle depending upon whether ASCEND) or HULL), respectively was the more recent command.

Determination of whether the vehicle motions being plotted are for the capsule or remaining hull is made by checking whether the value of the subroutine argument is 1, or 2, respectively. This us used to set the proper label and to compute vehicle length which are printed as part of the plot information.

The data arrays which are plotted are passed through the COMMON labelled INØUT. Scaling of the arrays is performed by Typagraph supplied software as is the drawing and labeling of the axes.

The subroutine also prints a table of the vehicle's general characteristics for identification purposes. The completion of the subroutine is followed by a return to the main program for additional monitor commands.

There are no error messages.

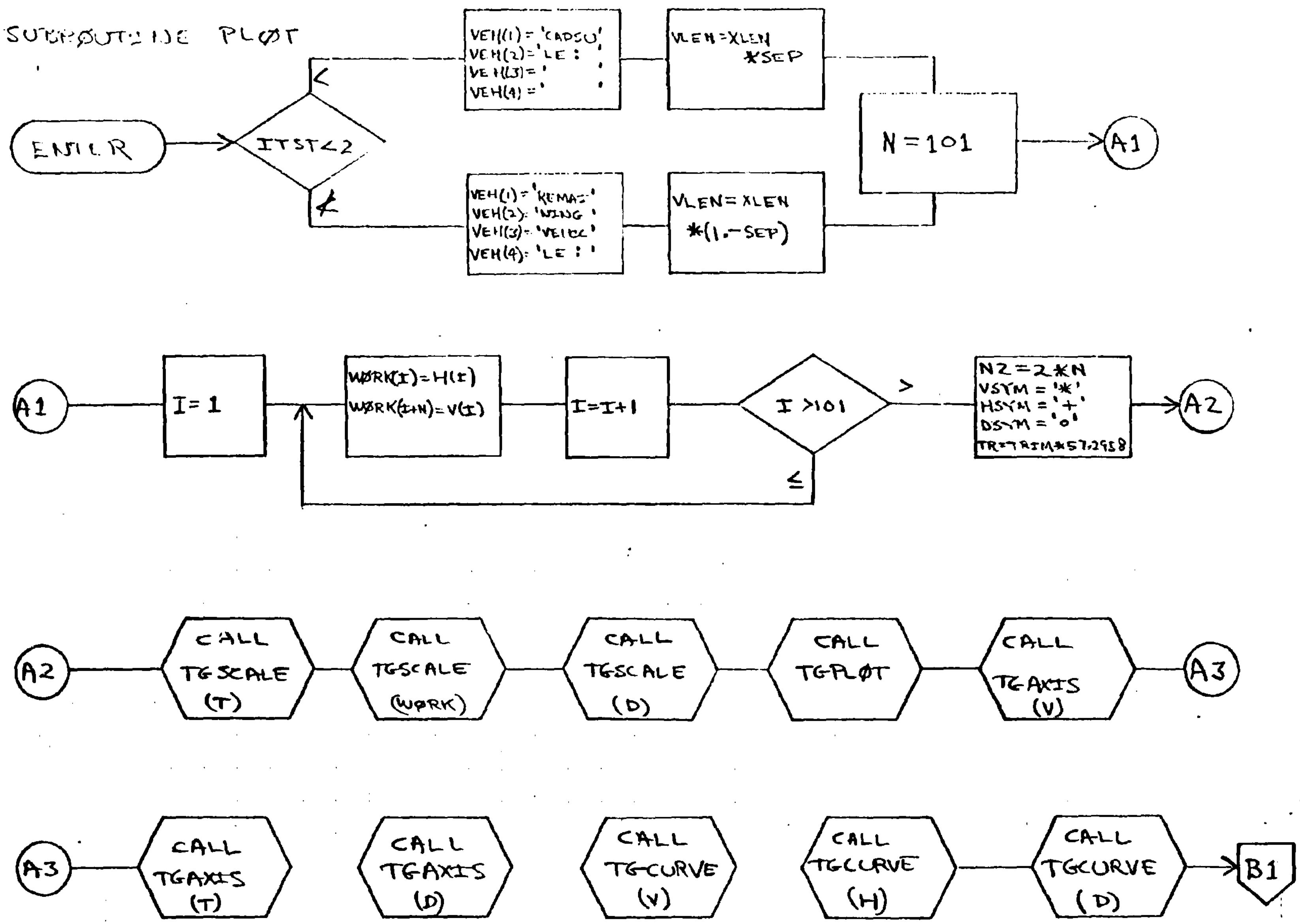
Input Variables

VTST	Capsule - remaining vehicle switch
XLEN	Intact vehicle length (ft.)
SEP	Separation point from nose (decimal fraction)
ELØD	Length to diameter ratio
TRIM	Initial trim angle (deg.)
XVEL	Initial horizontal velocity (ft./sec.)
YVEL	Initial vertical velocity (ft./sec.)
T	Time array, non-dimensional
D	Vehicle attitude array (deg.)
V	Vertical distance array, non-dimensional
H	Horizontal distance array, non-dimensional

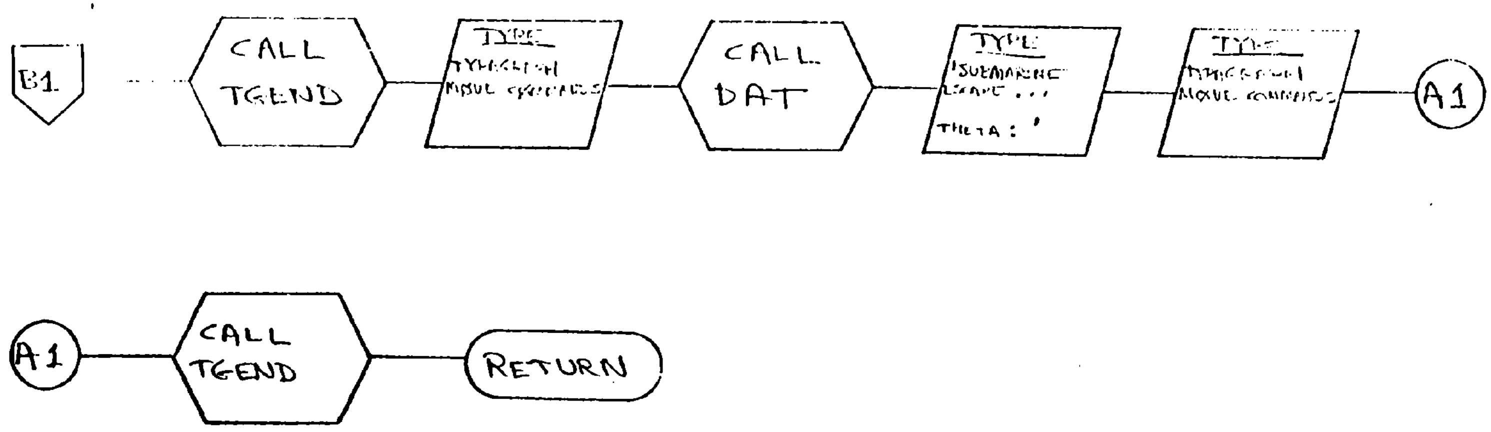
Output Variables

DAT	Date of simulation
XLEN	Intact vehicle length (ft.)
SEP	Separation point from nose (decimal fraction)
ELØD	Length to diameter ratio
TRIM	Initial trim angle (deg.)
XVEL	Initial horizontal velocity (ft./sec.)
YVEL	Initial vertical velocity (ft./sec.)
T	Time array, non-dimensional
D	Vehicle attitude array (deg.)
V	Vertical distance array, non-dimensional
H	Horizontal distance array, non-dimensional

IX-7.1 Flow Chart for Subroutine PLØT


SUBROUTINE PLØT
ENTER
ITST<2
VEH(1)='CADSU'
VEH(2)='LE :
VEH(3)='
VEH(4)='
VLEN=XLEN *SEP
VEH(1)='KEMA:'
VEH(2)='WING'
VEH(3)='VEIOC'
VEH(4)='LE :'
VLEN=XLEN *(1.-SEP)
N=101
A1
I=1
WØRK(I)=H(I)
WØRK(I+N)=V(I)
I=I+1
I >101
N2=2*N
VSYM='*'
HSYM='+'
DSYM='o'
TR=TRIM*57.2958
A2
CALL TGSCALE (T)
CALL TGSCALE (WØRK)
CALL TGSCALE (D)
CALL TGPLØT
CALL TGAXIS (V)
A3
CALL TGAXIS (T)
CALL TGAXIS (D)
CALL TGCURVE (V)
CALL TGCURVE (H)
CALL TGCURVE (D)
B1

Pg. 2


B1
CALL TGEND
TYPE
CALL DAT
TYPE
'SUBMARINE
THETA: '
TYPE
A1
A1
CALL TGEND
RETURN

IX-7.2 Listing of Subroutine PLØT

```
      SUBROUTINE PLOT(ITST,NPTS)
      COMMON/INPUT/XLEN,SEP,ELOD,TRIM,XVEL,YVEL,DVEL
      COMMON/INOUT/SKIP(27),T(101),D(101),V(101),H(101),EX(303)
      INTEGER TTITLE,DTITLE,HTITLE
      DIMENSION TTITLE(12),DTITLE(12),HTITLE(12),WORK(202)
      DIMENSION DAT(2),VEH(4)
      DATA(TTITLE(K),K=1,12)/'NON-DIMENSIONAL TIME',
     *  '                                            '/
      DATA( DTITLE(K),K=1,12)/'VEHICLE ATTITUDE IN DEGREES   ',
     * '          ','                 '/
      DATA(HTITLE(K),K=1,12)/'DISTANCE FROM SEPARATION POINT     ',
     *  '                        '/
      IF(ITST-1)1,1,2
 1    VEH(1)='CAPSU'
      VEH(2)='LE : '
      VEH(3)='     '
      VEH(4)='     '
      VLEN=XLEN*SEP
      GOTO 3
 2    VEH(1)='REMAI'
      VEH(2)='NING '
      VEH(3)='VEHIC'
      VEH(4)='LE : '
      VLEN=XLEN*(1.-SEP)
 3    CONTINUE
C "N"= LUMBER GF POINTS TO BE PLOTTED
      N= NPTS
       DO 20 I=1,N
       WORK(I)=H(I)
 20    WORK(I+N)=V(I)
      N2=2*N
      TR=TRIM*57.2958
C ASSIGNMENT OF ARRAY SYMBOLS  *, Δ, •
      VSYM='*'
      HSYM='+'
      DSYM='.'
C SCALE ALL ARRAYS
      CALL TGSCALE(T,N,6.,TMIN,TSCALE,1)
       CALL TGSCALE(WORK,N2,7.75,VMIN,VSCALE,1)
      CALL TGSCALE(D,N,6.,DMIN,DSCALE,1)
C SET INITIAL ORIGIN FOR PLOTTING AXES
      CALL TGPLOT(9.0,0)
C DRAW LEFTMOST VERTICAL AXIS
      CALL TGAXIS(0.0,0.0,7.75,VMIN,VSCALE,1,10,2,HTITLE)
      CALL TGAXIS(0.0,0.0,6.0,TMIN,TSCALE,4,-10,2,TTITLE)
       CALL TGAXIS(6.0,0.0,6.0,DMIN,DSCALE,2,-10,2,DTITLE)
```

```
C  PLOT HORIZ DISTANCE
       CALL TGCURVE(T,V,N,TMIN,VMIN,TSCALE,VSCALE,VSYM,1 )
C  PLOT VERT. DISTANCE
       CALL TGCURVE(T,H,N,TMIN,VMIN,TSCALE,VSCALE,HSYM,1)
C  PLOT ATTITUDE
       CALL TGCURVE(T,D,N,TMIN,DMIN,TSCALE,DSCALE,DSYM,1)
       CALL TGEND
       TYPE 30
 30     FORMAT('+'IO? ???????????????????????????????????'D40 ')
       CALL DATE(DAT)
       TYPE 40,DAT,VEH,VLEN,ELOD,SEP,TR,XVEL,YVEL
 40    FORMAT(1HO,T17,'SUBMARINE ESCAPE CAPSULE SIMULATION',/,
     * T25,'ONR , '2A5,/T17,'CHARACTERISTICS OF '4A5,/,
     * T28,'LENGTH (FT) : ',F8.2,/,T15,'LENGTH TO DIAMETER RATIO : ',
     * F8.2,/,T14,'SEPARATION PLANE LOCATION : ',F8.2,/,
     * T15,'INITIAL TRIM ANGLE (DEG) : ',F8.2,/,
     * T12,'INITIAL X VELOCITY (FT/SEC) : ',F8.2/,
     * T12,'INITIAL Y VELOCITY (FT/SEC) : ',F8.2,/,
     * T36,'X/L :    +',/,T36,'Y/L :    *',/,T34,'THETA :    .')
       TYPE 50
 50    FORMAT('+'JO? ?????????????????????????????????????')
       CALL TGEND
       RETURN
       END
```

IX-8. Subroutine PRNT

The data arrays describing the vehicle's (capsule or remaining vehicle) motions following separation may be plotted (cf. Subroutine PLØT) or printed out in tabular form (Subroutine PRNT). PRNT allows the user to sample two to fourteen points of each array (the first and last points of the array are always included) for inspection of their numerical values. The user calls the subroutine from the monitor by typing PRINT). The subroutine responds by requesting the number of points the user wants to print and uses that number to calculate an approximately equally spaced sample for tabulation. Because of integer truncation and other physical limitations, the algorithm that selects the points to be plotted begins to lose its accuracy if more than fourteen points are requested. No restrictions, however, are placed on any number requested between 2 and 101. The error message "ERROR, NUMBER OF POINTS MUST FALL WITHIN THE RANGE [2, 101]. TRY AGAIN." is given if the user tries to print less than 2 or more than 101 points. The user is then again asked "HOW MANY POINTS DO YOU WISH TO PRINT 2 TO 14 --".

Non-dimensional results are first printed out as a table. Then the values are dimensionalized and printed out as a second table.

Input Variables

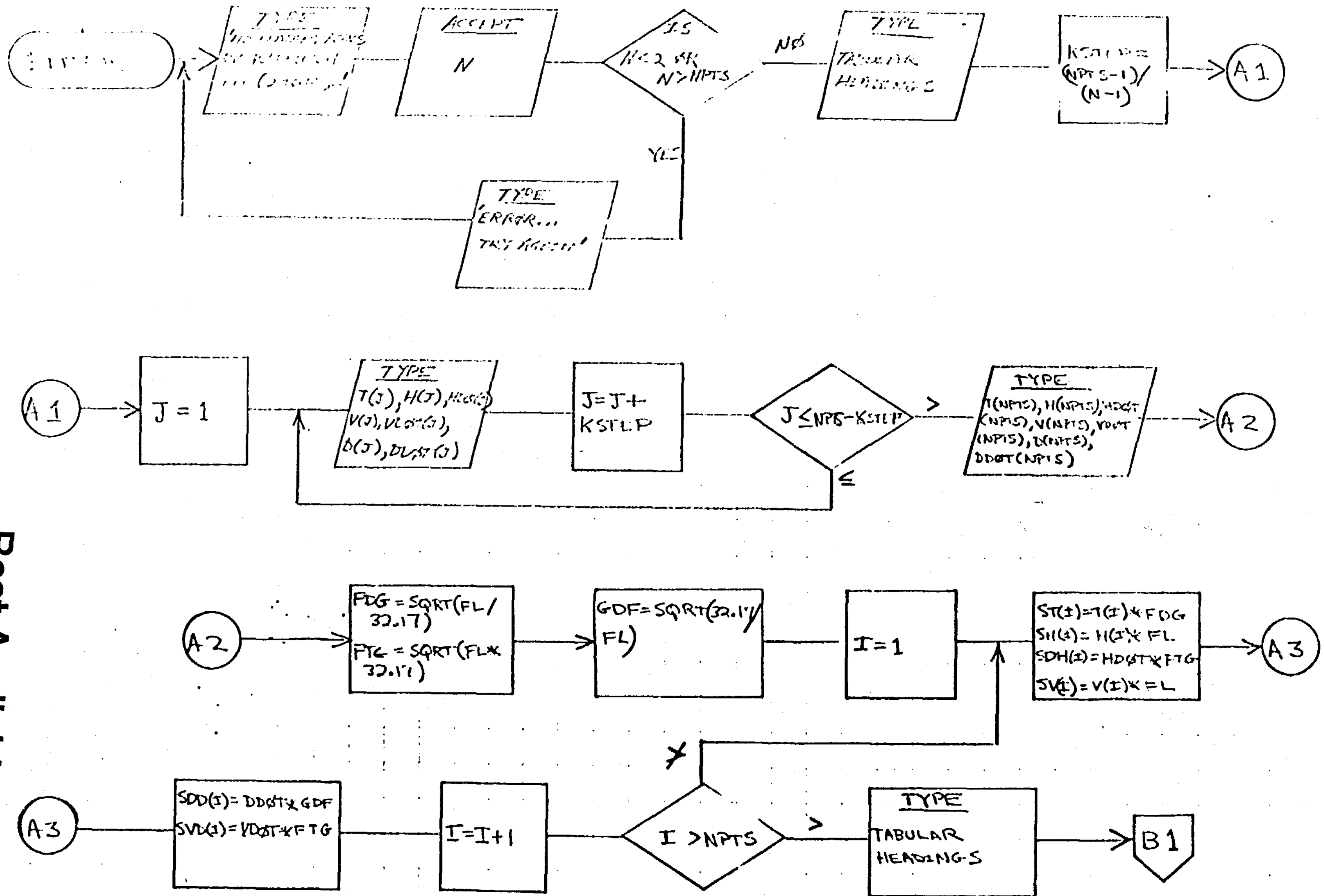
FL	Length of vehicle (ft.)
T	Time array, non-dimensional
D	Vehicle attitude array (deg.)
V	Vertical distance array, non-dimensional
H	Horizontal distance array, non-dimensional
DDØT	Angular velocity array (deg./non-dim. time)
VDØT	Vertical velocity array, non-dimensional
HDØT	Horizontal velocity array, non-dimensional
N	Number of points to be printed
NPTS	Number of points in each array

Output Variables

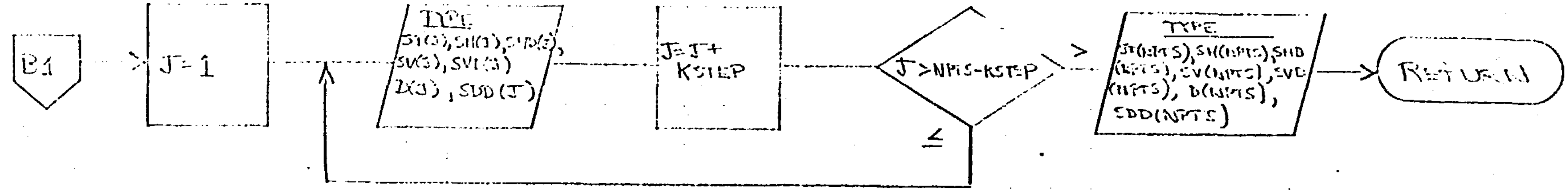
T	Time array, non-dimensional
D	Vehicle attitude array (deg.)
V	Vertical distance array, non-dimensional
H	Horizontal distance array, non-dimensional
DDØT	Angular velocity array (deg./sec.)
VDØT	Vertical velocity array, non-dimensional)
HDØT	Horizontal velocity array, non-dimensional
ST	Time array (sec.)
SV	Vertical distance array (ft.)
SH	Horizontal distance array (ft.)
SVD	Vertical velocity (ft./sec.)
SHD	Horizontal velocity (ft./sec.)
SDD	Angular velocity (deg./sec.)

IX-8.1 Flow Chart for Subroutine PRNT

SUBROUTINE PRINT



PRNT



Best Available Copy

IX-8.2 Listing of Subroutine PRNT

```
      SUBROUTINE PRNT(FL,NPTS)
      DIMENSION ST(101),SH(101),SHD(101),SV(101),
     *SVD(101),SDD(101)
      COMMON/INPUT/FLOA,SEP,FLDRAT,TRIM,XVEL,YVEL,DVEL
      COMMON/INOUT/SKIP(27),T(101),D(101),V(101),H(101),
     *DDOT(101),VDOT(101),HDOT(101)
1      TYPE 5
      ACCEPT 6,N
      IF(N.LT.2.OR.N.GT.NPTS)GO TO 100
      TYPE 10
      TYPE 20
10     FORMAT(///,1H-,T24,'NON-DIMENSIONAL RESULTS')
20     FORMAT(1H ,T4,'TIME',T16,'HORIZONTAL',T37,'VERTICAL',
     *T52,'ATTITUDE',T62,'ANGULAR',/,T12,'DISTANCE',T22,
     *'VELOCITY',T32,'DISTANCE',T42,'VELOCITY',T52,'DEGREES',
     *T62,'VELOCITY')
      TYPE 30
30    FORMAT(1H ,T3,'-------',T12,'--------',T22,'--------',T32,
     *'--------',T42,'--------',T52,'--------',T62,'--------')
5     FORMAT(' HOW MANY POINTS DO YOU WISH TO PRINT (2 TO 14) -- ',
     *$)
6     FORMAT(I)
      KSTEP=(NPTS-1)/(N-1)
      DO 50 J=1,NPTS-KSTEP,KSTEP
50    TYPE 40,T(J),H(J),HDOT(J),V(J),VDOT(J),D(J),DDOT(J)
      TYPE 40,T(NPTS),H(NPTS),HDOT(NPTS),V(NPTS),VDOT(NPTS),
     *D(NPTS),DDOT(NPTS)
40    FORMAT(1H ,7(F8.2,2X))
C  CONVERSION TO DIMENSIONALIZED VALUES IS COMPUTED NEXT.
C
C
      FDG=SQRT(FL/32.17)
      FTG=SQRT(FL*32.17)
      GDF=SQRT(32.17/FL)
      DO 60 I=1,NPTS
      ST(I)=T(I)*FDG
      SH(I)=H(I)*FL
      SHD(I)=HDOT(I)*FTG
      SV(I)=V(I)*FL
      SDD(I)=DDOT(I)*GDF
60    SVD(I)=VDOT(I)*FTG
C
C
C
```

```
      TYPE 70
70    FORMAT(1H-,T24,'DIMENSIONALIZED RESULTS')
      TYPE 20
      TYPE 80
80    FORMAT(1H ,T3,'SECONDS',T14,'FEET',T23,'FT/SEC',T34,'FEET',T43,
     *'FT/SEC',T63,'DEG/SEC')
      TYPE 30
     DO 90 J=1,NPTS-KSTEP,KSTEP
90    TYPE 40,ST(J),SH(J),SHD(J),SV(J),SVD(J),D(J),SDD(J)
      TYPE 40,ST(NPTS),SH(NPTS),SHD(NPTS),SV(NPTS),SVD(NPTS),
     *D(NPTS),SDD(NPTS)
      GO TO 120
100    TYPE 110,NPTS
110    FORMAT(' ERROR, NUMBER OF POINTS MUST FALL WITHIN',/
     * ' THE RANGE [2,',I3,'].  TRY AGAIN.',/)
      GO TO 1
120   TYPE 130
130   FORMAT(//)
      RETURN
      END
```

IX-9. Subroutine PDATA

In order to provide the user with a current status of what he has done, the command CHECK} has been incorporated into the monitor. This command calls subroutine PDATA which types out the status of the simulation. It does this by testing the status variables of the pertinent subroutines which have been passed as subroutine arguments. If the test is affirmative, a message is typed out notifying the user that the subroutine in question has been exercised. The general characteristics of the intact vehicle are next listed and the user has the option of inspecting the current weight and/or volume distribution.

Upon completion of the status report, control returns to the monitor.

There are no error messages.

Input Variables

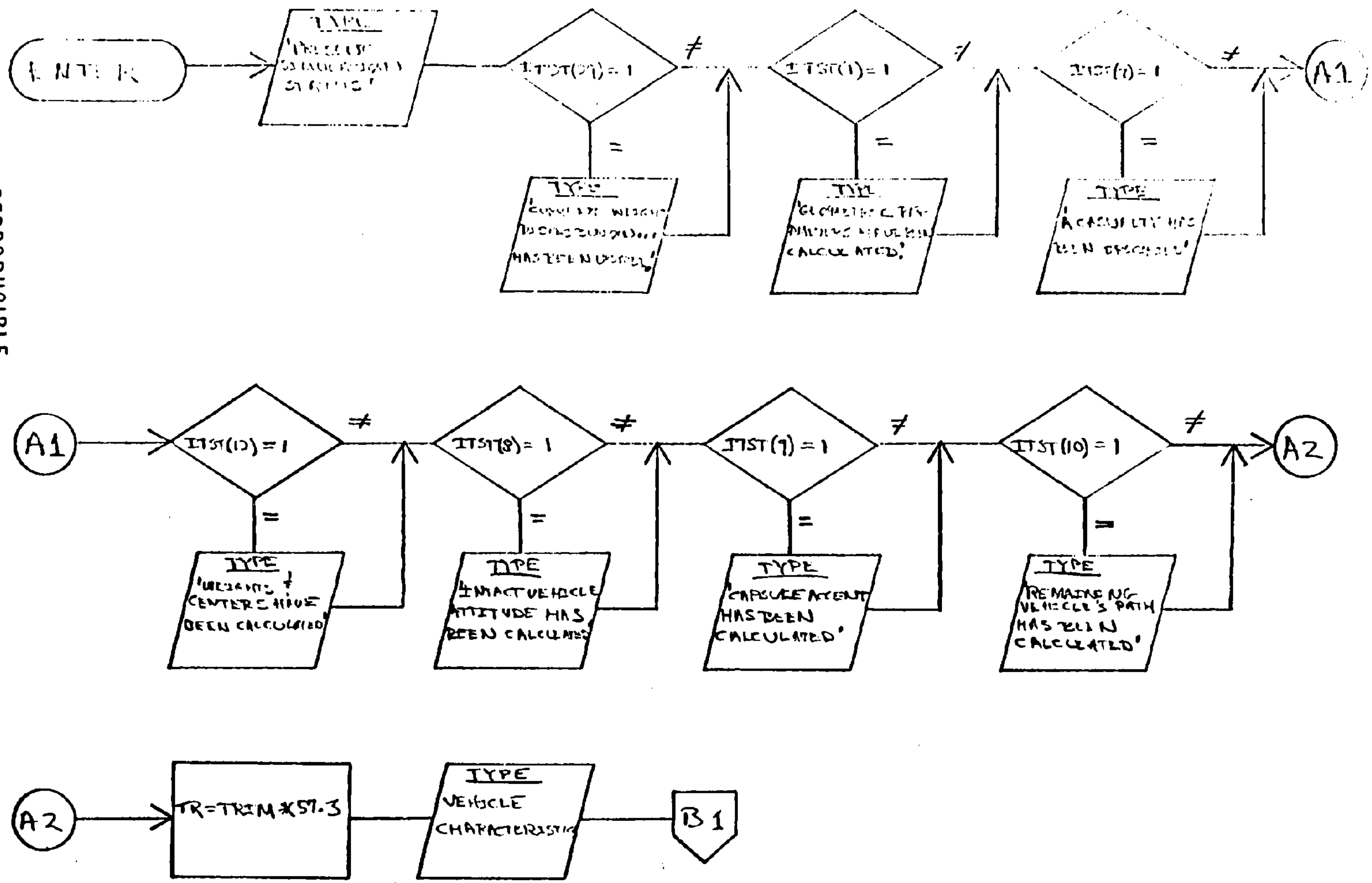
ITST	Subroutine status check array
FLØØD	Floodable spaces distribution array (ft.^{3}, dec. frac., dec. frac.)
WEIGHT	Weight distribution array (lbs., dec. frac., dec. frac.)
NUMWGT	Number of weights in WEIGHT array
NUMFLD	Number of spaces in FLØØD array
CAPT	Capsule shell thickness (in.)
HULLT	Hull shell thickness (in.)
FLEN	Intact vehicle length (ft.)
SEP	Separation point from nose (decimal fraction)
FLDRAT	Length to dismeter ratio
TRIM	Initial trim angle (radians)
VELX	Initial horizontal velocity (ft./sec.)
VELY	Initial vertical velocity (ft./sec.)
DINIT	Initial depth (ft.)

Output Variables

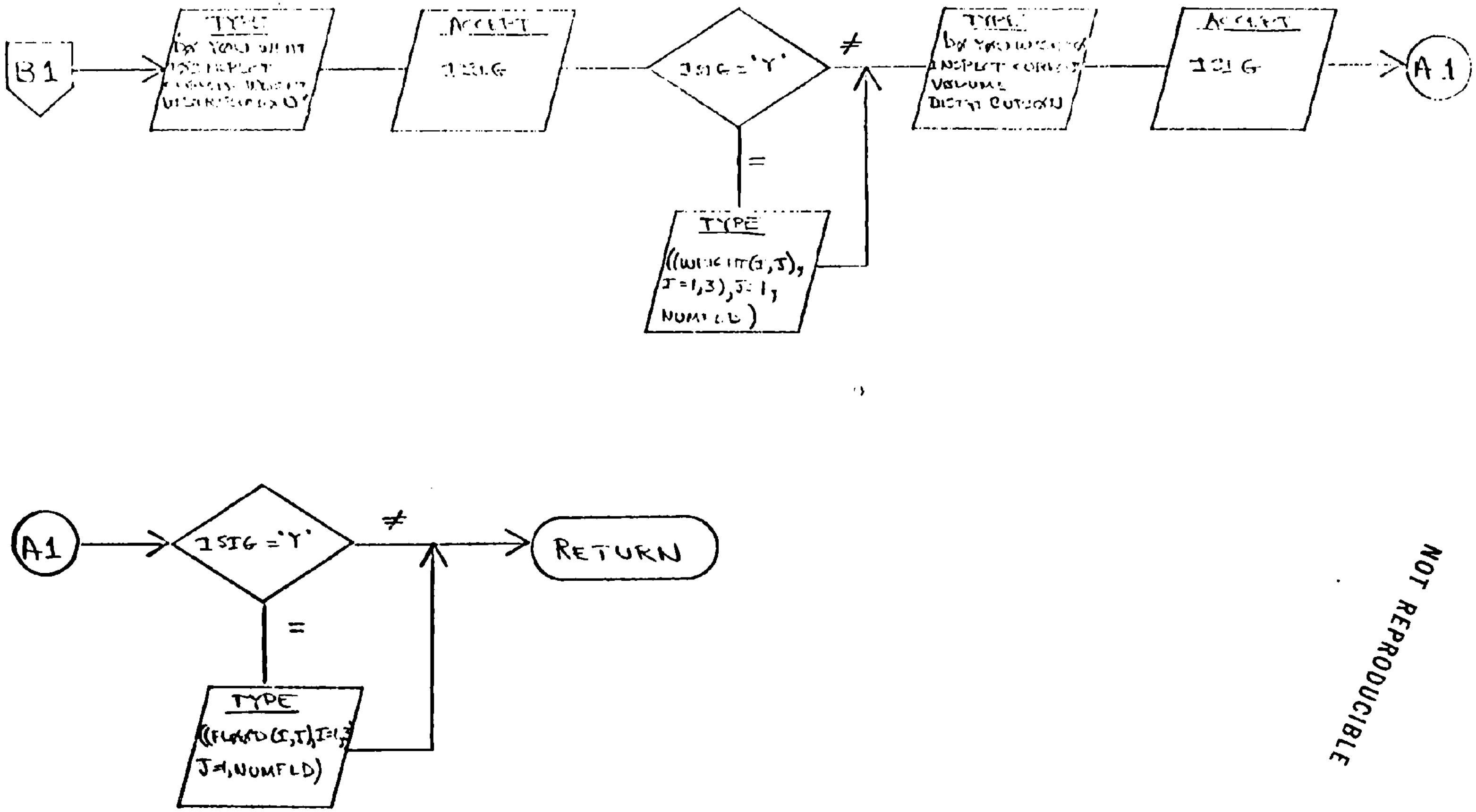
FLDRAT	Length to diameter ratio
FLEN	Intact vehicle length (ft.)
SEP	Separation point from nose (dec. frac.)
CAPT	Capsule shell thickness (in.)
HULLT	Hull shell thickness (in.)
VELX	Initial horizontal velocity (ft./sec.)
VELY	Initial vertical velocity (ft./sec.)
DINIT	Initial depth (ft.)
TR	Initial trim angle (deg.)
WEIGHT	Weight distribution array (lbs., dec. frac., dec. frac.)
FLØØD	Floodable spaces distribution array (lbs., dec. frac., dec. frac.)

IX-9.1 Flow Chart for Subroutine PDATA

SUBROUTINE PDATA



NOT REPRODUCIBLE


B1
TYPE
ACCEPT
ISIG
ISIG = 'Y'
=
≠
TYPE
ACCEPT
ISIG
A1
A1
ISIG = 'Y'
=
≠
TYPE
J=1,NUMFLD)
RETURN

IX-9.2 Listing of Subroutine PDATA

```
      SUBROUTINE PDATA(ITST,FLOOD,WEIGHT,NUMWGT,NUMFLD,CAPT,HULLT,
     * DINIT,FLCGIN,VCGITV,FLCGCP,VCGCAP)
      COMMON /INPUT/FLEN,SEP,FLDRAT,TRIM,VELX,VELY,VELD
      DIMENSION ITST(30),FLOOD(3,100),WEIGHT(3,100)
      TYPE 10
10    FORMAT(//,15X,'PRESENT SIMULATION STATUS',/)
      IF(ITST(29).EQ.1)TYPE 202
      IF(ITST(1).EQ.1)TYPE 201
      IF(ITST(7).EQ.1)TYPE 207
      IF(ITST(12).EQ.1)TYPE 212
      IF(ITST(8).EQ.1)TYPE 208
      IF(ITST(9).EQ.1)TYPE 209
      IF(ITST(10).EQ.1)TYPE 210
      TR=TRIM*57.2958
      TYPE 110,FLDRAT,FLEN,SEP,CAPT,HULLT,VELX,VELY,DINIT,TR
     * ,FLCGIN,VCGITV,FLCGCP,VCGCAP
      TYPE 111
      ACCEPT 50,ISIG
      IF(ISIG.EQ.'N')GO TO 15
      DO 11 I=1,NUMWGT
11    WEIGHT(1,I)=WEIGHT(1,I)/2240.
      TYPE 112,((WEIGHT(I,J),I=1,3),J=1,NUMWGT)
      DO 12 I=1,NUMWGT
12    WEIGHT(1,I)=WEIGHT(1,I)*2240.
15       TYPE 113
      ACCEPT 50,ISIG
      IF(ISIG.EQ.'Y')TYPE 114,((FLOOD(I,J),I=1,3),J=1,NUMFLD)
      RETURN
110   FORMAT(/' LENGTH/DIAMETER RATIO                ='  F10.2,/
     *         ' TOTAL VEHICLE LENGTH(FT)             ='  F10.2,/
     *         ' SEPARATION PLANE LOCATION            ='  F10.2,/
     *         ' CAPSULE SHELL THICKNESS(IN.)         ='  F10.2,/
     *         ' HULL SHELL THICKNESS(IN.)            ='  F10.2,/
     *         ' INITIAL HORIZONTAL VELOCITY          ='  F10.2,/
     *         ' INITIAL  VERTICAL VELOCITY           ='  F10.2,/
     *         ' INITIAL DEPTH(FT)                    ='  F10.2,/
     *          ' INITIAL TRIM ANGLE(DEGREES)          ='  F10.2,/
     *         ' LCG INTACT VEHICLE                   = ' F10.3/
     *         ' VCG INTACT VEHICLE                   = 'F10.3/
     *          ' LCG CAPSULE                          = ' F10.3/
     *         ' VCG CAPSULE                          = ' F10.3/
     *   )
```

```
111     FORMAT(/' DO YOU WISH TO INSPECT THE CURRENT WEIGHT'
    *   ' DISTRIBUTION'/' TYPE Y OR N - '$)
113     FORMAT(/' DO YOU WISH TO INSPECT THE CURRENT VOLUME'
    *   ' DISTRIBUTION?'/' TYPE Y OR N - '$)
112     FORMAT(/' CURRENT WEIGHT DISTRIBUTION IS:'//,(5X,3F12.2))
114     FORMAT(/' CURRENT VOLUME DISTRIBUTION IS :'//,(5X,3F12.2))
50       FORMAT(A1)
202     FORMAT (/' CURRENT WEIGHT DISTRIBUTION OR VOLUME DIS'
    *   'TRIBUTION'/ ' HAS BEEN UPDATED')
201     FORMAT(/' GEOMETRICAL PARAMETERS HAVE BEEN CALCULATED')
207     FORMAT(/' A CASUALTY HAS BEEN SPECIFIED')
212     FORMAT(/' WEIGHTS AND CENTERS HAVE BEEN CALCULATED')
208     FORMAT(/' INTACT VEHICLE ATTITUDE HAS BEEN CALCULATED')
209     FORMAT(/' CAPSULE ACCENT HAS BEEN CALCULATED')
210     FORMAT(/' REMAINING VEHICLES PATH HAS BEEN CALCULATED')
        END
```

IX-10. Subroutine ADDØN

This subroutine updates the weight and volume distribution files maintained on disk. ADDON is not called directly by a command; however, the option to add or update distributions is provided when a distribution is input from the keyboard or when a distribution has been updated. Distributions may be added to the file or in the case of an updated distribution, the old distribution on file is replaced by the updated version. The user must input a "distribution identifier" which is 5 characters long; it may be alphanumeric and must be unique in the file. If the user inputs an identifier for which there already exsists distribution, he may not input a new identifier for his distribution. Once this operation is completed the message "DISTRIBUTION HAS BEEN STORED FOR FUTURE USE" is output and control returns to the monitor.

There are no error messages.

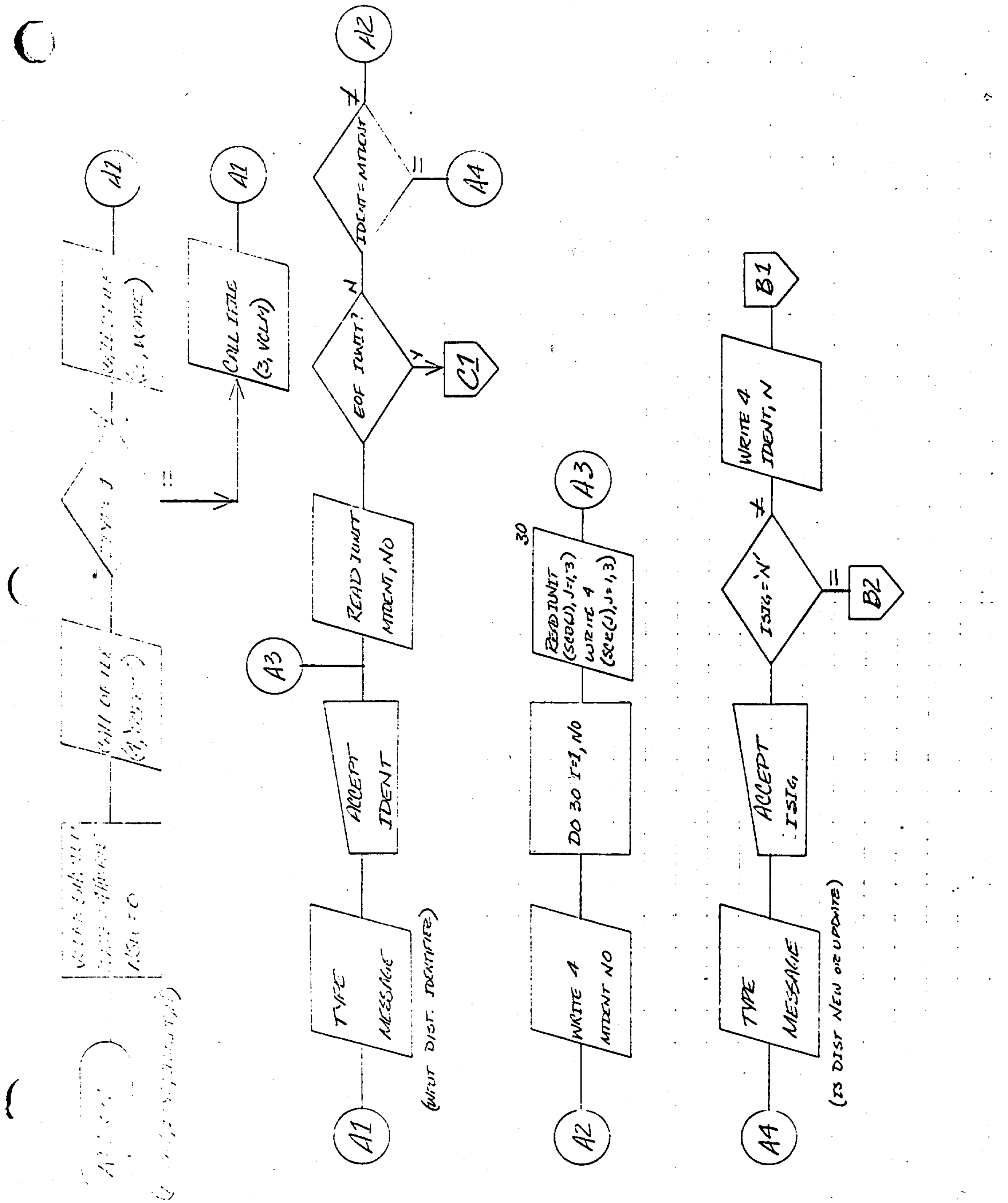
Input Variables

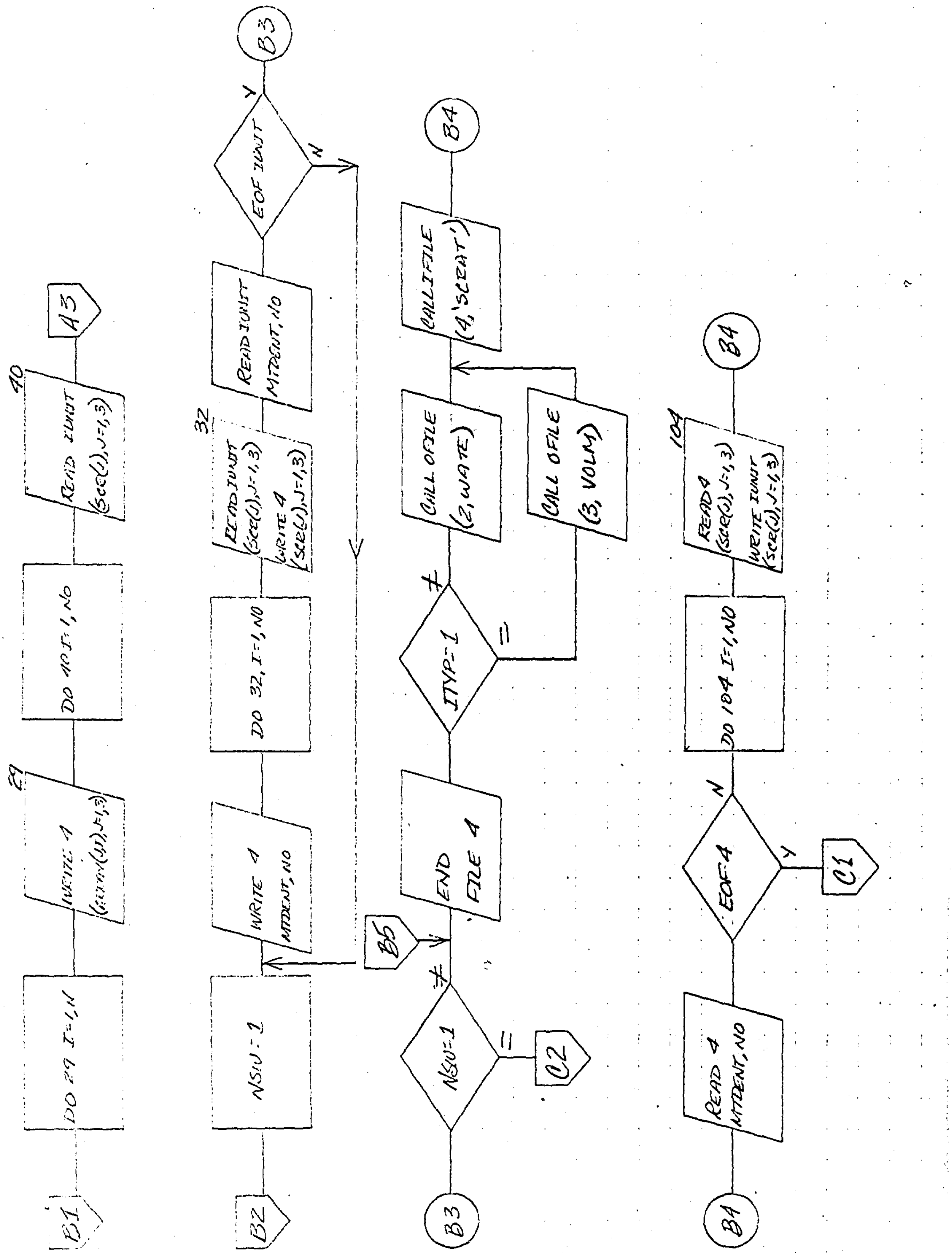
ARRAY	Weight or volume array to be output on file
ITYP	Specifies that array contains weights or volumes
IUNIT	Specifies weight or volume file number
N	Number of triplets

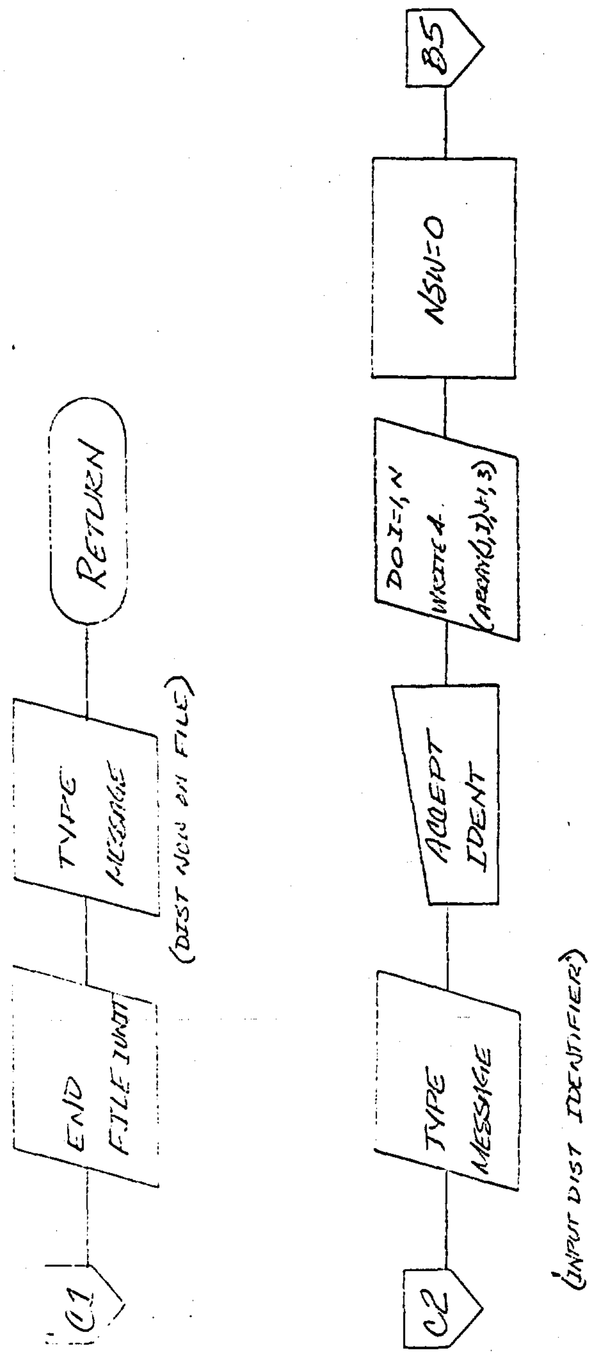
Output Variables

None

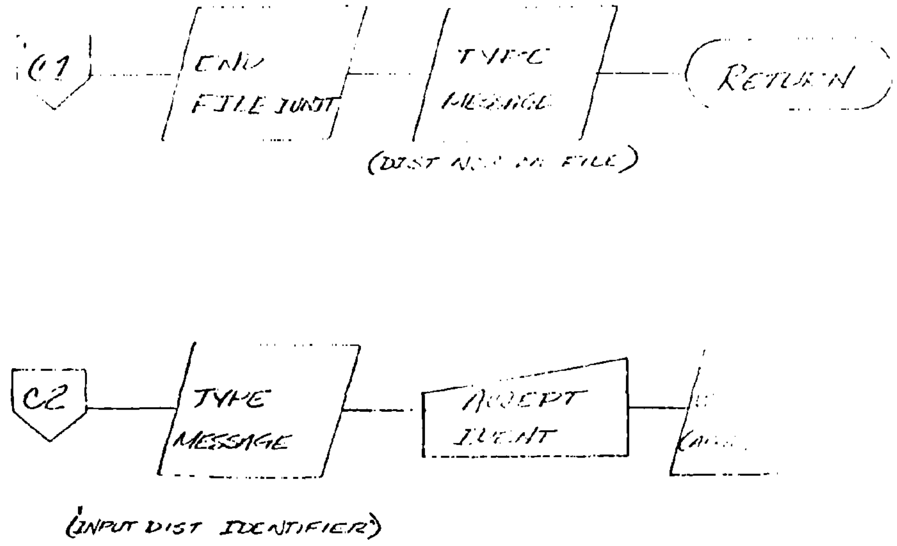
IX-10.1 Flow Chart for Subroutine ADDØN


A1
TYPE MESSAGE
(INPUT DIST. IDENTIFIER)
ACCEPT IDENT
A3
READ IUNIT MIDENT, NO
EOF IUNIT?
Y
C1
N
IDENT = MIDENT
=
A4
≠
A2
CALL TITLE (3, VCLM)
A1
A2
WRITE 4 MIDENT NO
DO 30 I=1, NO
30
READ IUNIT (SCR(J), J=1,3) WRITE 4 (SCR(J), J=1,3)
A3
A4
TYPE MESSAGE
(IS DIST NEW OR UPDATE)
ACCEPT ISIG
ISIG = 'N'
=
B2
≠
WRITE 4 IDENT, N
B1


B1
DO 29 I=1,N
29
WRITE 4 (...(I,J),J=1,3)
DO 40 J=1,NO
40
READ IUNIT (SCR(J),J=1,3)
A3
B2
NSW=1
WRITE 4 MTDENT,NO
DO 32, I=1,NO
32
READ IUNIT (SCR(J),J=1,3) WRITE 4 (SCR(J),J=1,3)
READ IUNIT MTDENT,NO
EOF IUNIT
Y
N
B3
B3
NSW=1
=
≠
C2
B5
END FILE 4
ITYP=1
≠
=
CALL OFILE (2,WATE)
CALL OFILE (3,VOLM)
CALL IFILE (4,'SCRAT')
B4
B4
READ 4 MTDENT,NO
EOF 4
Y
N
C1
DO 104 I=1,NO
104
READ 4 (SCR(J),J=1,3) WRITE IUNIT (SCR(J),J=1,3)
B4


C1
END
FILE UNIT
TYPE
MESSAGE
(DIST NOW ON FILE)
RETURN
C2
TYPE
MESSAGE
(INPUT DIST IDENTIFIER)
ACCEPT
IDENT
DO I=1,N
WRITE 4
(ARRAY(I,J),J=1,3)
NSW=0
B5

IX-10.2 Listing of Subroutine ADDØN


C1
TYPE
RETURN
C2
TYPE
MESSAGE
(INPUT DIST IDENTIFIER)

IX-10.2 Listing of Subroutine ADDØN

```
TYPE ADDON.FOR
      SUBROUTINE ADDON(ARRAY,ITYP,IUNIT,N)
      DIMENSION ARRAY(3,100),SCR(3)
C*******ROUTINE TO ADD DISTRIBUTION TO DATA FILE.
      VOLM=5HVOLUM
      WATE=4HWATE
      NSW=0
      ISIG=' '
      CALL OFILE(22,'SCRAT')
      IF(ITYP.EQ.1)GO TO 10
C*****.*IITYP=1 ADD TO VOLUMES, ITYPE=0 ADD TO WEIGHTS.
20    CALL IFILE(21,WATE)
      GO TO 15
10    CALL IFILE(23,VOLM)
C*******FILES OPEN 20=WEIGHT FILE, 21=VVOLUME FILE, 4=SCRATCH
15      TYPE 50
      ACCEPT 51,IDENT
25      READ(IUNIT,52)MTDENT,NO
      IF(EOFC(IUNIT)) GO TO 36
      IF(MTDENT.EQ.IDENT)GO TO 26
      WRITE(22,52)MTDENT,NO
      DO 30 I=1,NO
      READ(IUNIT,53)(SCR(J),J=1,3)
30    WRITE(22,53)(SCR(J),J=1,3)
      GO TO 25
27    END FILE 22
      GO TO 101
26    TYPE 54
54    FORMAT(' IS DISTRIBUTION NEW OR AN UPDATE ?'/
     * ' TYPE N OR U -- '$)
      ACCEPT 55, ISIG
55    FORMAT(A1)
      IF(ISIG.EQ.'N')GO TO 28
C*******WRITE UPDDATEPDISTRIBUTION
      WRITE(22,52) IDENT,N
      DO 29 I=1,N
29    WRITE(22,53)(ARRAY(J,I),J=1,3)
C********SKIP OLDD DISTRIBUTION.
      DO 40 I=1,NO
40    READ(IUNIT,53)(SCR(J),J=1,3)
      GO TO 25
28    NSW=1
31    WRITE(22,52)MTDENT,NO
      DO 32 I=1,NO
33    READ(IUNIT,53)(SCR(J),J=1,3)
32      WRITE(22,53)(SCR(J),J=1,3)
      READ(IUNIT,52)MTDENT,NO
      IF (EOFC(IUNIT))GO TO 100
      GO TO 31
```

```
 100      IF(NSW.EQ.1)GO TO 34
 101     END FILE 22
         IF(ITYP.EQ.1)GO TO 39
         CALL OFILE(21,WATE)
         GO TO 103
 39      CALL OFILE(23,VOLM)
 103     CALL IFILE(22,'SCRAT')
C*******START COPY FROM SCRATCH FILE TO ROIG INAL IUNIT.
105         READ(22,52)MTDENT,NO
         IF(EOFC(22))GO TO 102
         WRITE (IUNIT,52)MTDENT,NO
         DO 104 I=1,NO
           READ(22,53)(SCR(J),J=1,3)
104          WRITE(IUNIT,53)(SCR(J),J=1,3)
         GO TO  105
 102     END FILE IUNIT
         TYPE 57
 57      FORMAT(' DISTRIBUTION HAS BEEN STORED FOR FUTURE USE'/)
          RETURN
C*******WRITE NEW DISTRIBUTION AT END OF LOG. 4.
34        TYPE 56
 56      FORMAT(' PLEASE INPUT  IDENTIFIER OF NEW DISTRIBUTION'/
     *   ' 5 CHARACTERS ALLOWED -- '$)
         ACCEPT 51,IDENT
 36      IF(ISIG.EQ.'U')GO TO 101
         WRITE (22,52)IDENT,N
         DO 35 I=1,N
 35      WRITE(22,53)(ARRAY(J,I),J=1,3)
         NSW=0
         GO TO 101
 50      FORMAT(' ENTER SHIP "IDENTIFIER", (5 CHARACTERS)  -- '$)
 51      FORMAT(A5)
 52      FORMAT(A5,I3)
 53      FORMAT(3F)
         END
```

IX-11 Subroutine UPDATE

This routine is called in response to the command UPDATE). The subroutine provides the user the option of updating either the weight or volume distribution once it is in storage. The user may at his option change, delete, or add a weight or volume and its centers. Depending upon the operation the user must input the old triplet that is to be deleted or changed and he must also input the new triplet if it is to be changed or added. The user must take caution to remember that the weights must be input in pounds for this routine. Once the distribution is updated, the user is given the option of storing the distribution for future use. Program control then returns to the monitor.

Error Messages

1) "WEIGHT OR VOLUME AND POSITION CANNOT BE FOUND" - The user has attempted to change or delete a weight or volume which does not exist in the distribution or which he may have input incorrectly. (Note: Weights must be expressed in pounds.)

2) "DETECT ERROR IN INPUT TRY AGAIN" - The command input by the user for changing, adding, or deleting is incorrect.

Input Variables

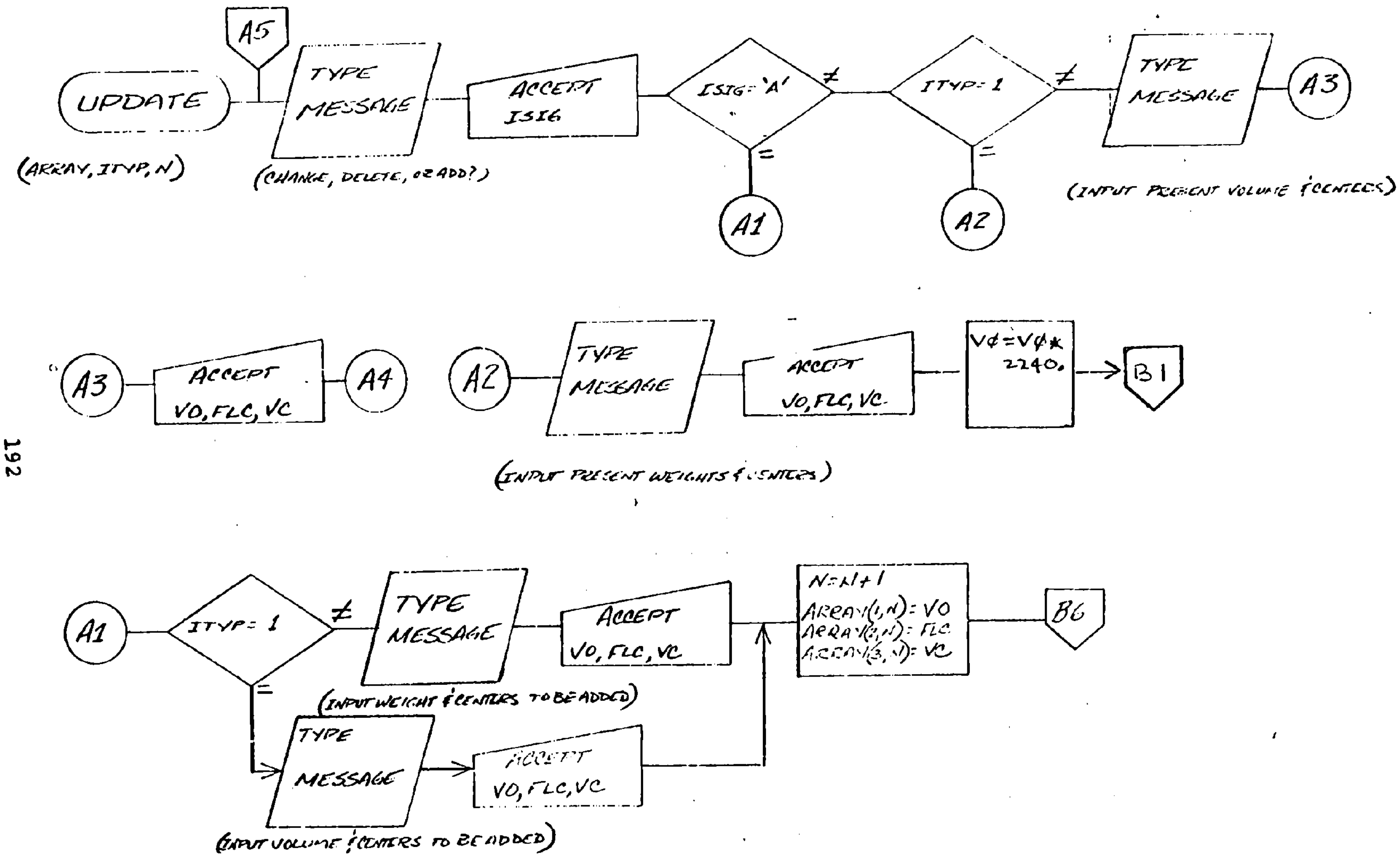
ARRAY	Weight or volume array to be updated
ITYP	Specifies whether weights or volumes are to be updated.
N	Number of weight or volume triplets in the distribution

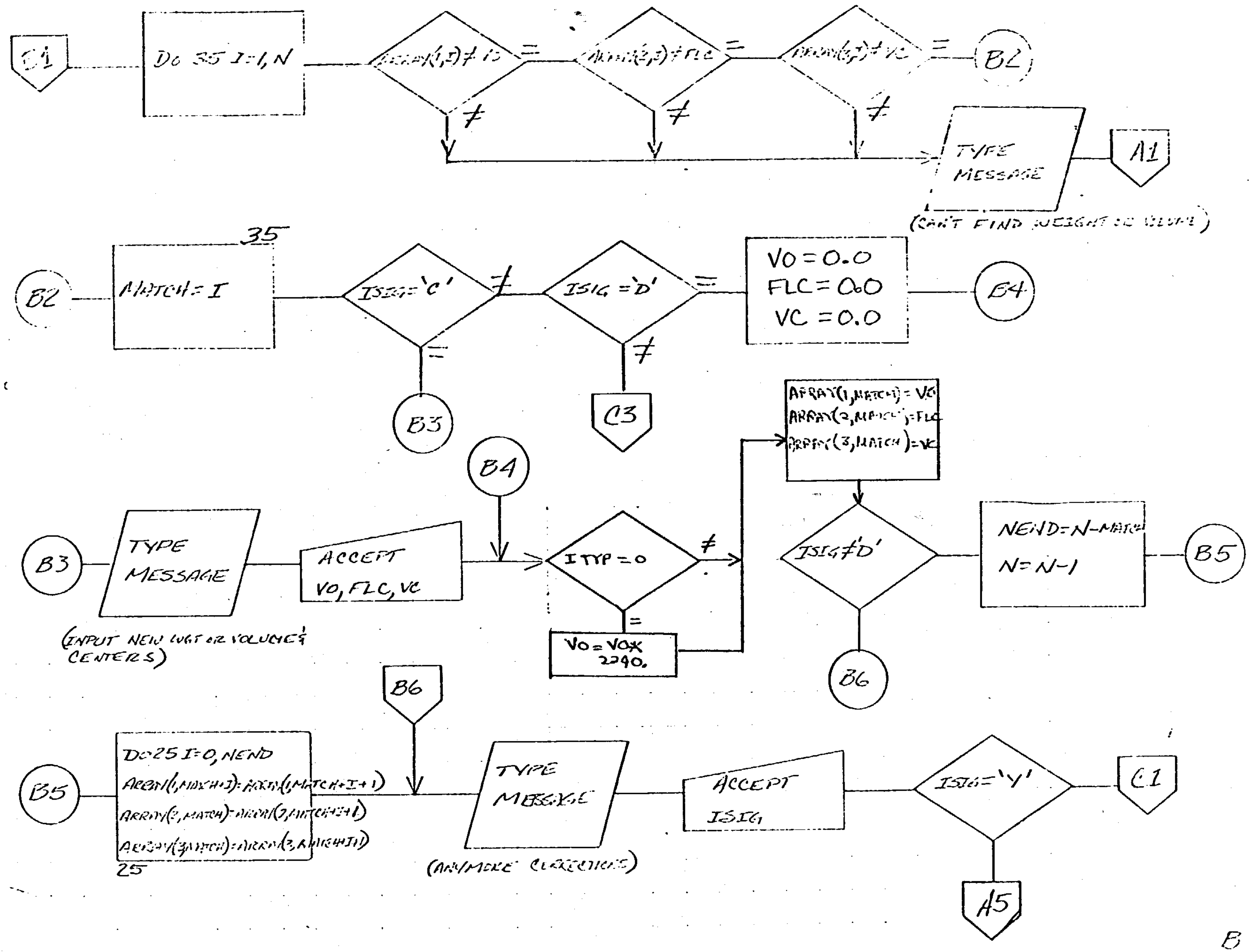
Output Variables

ARRAY	Update array

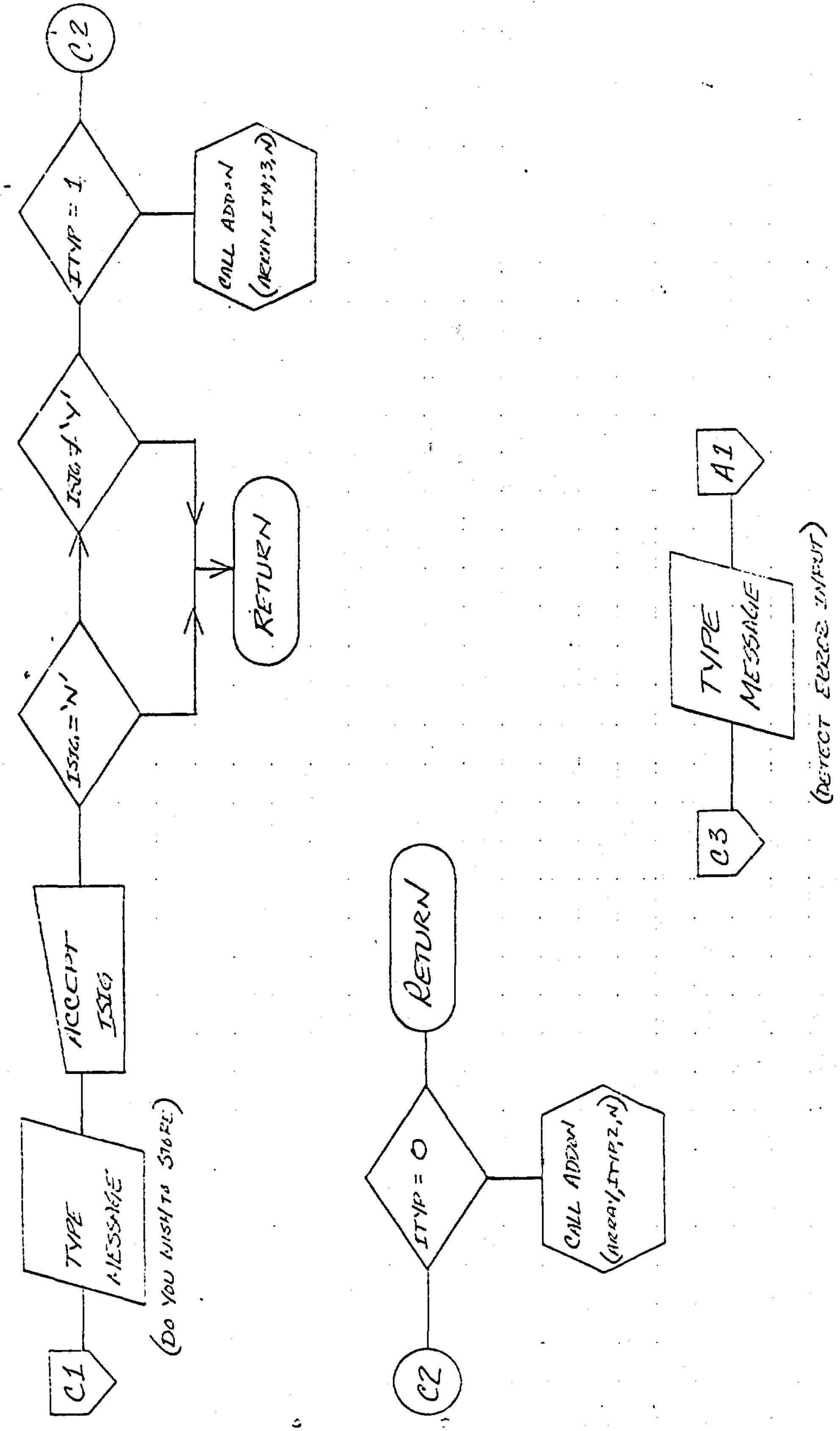
IX-11.1 Flow Chart for Subroutine UPDATE

SUBROUTINE UPDATE





C1
DO 35 I=1,N
≠
=
TYPE MESSAGE
(CAN'T FIND WEIGHT OR VOLUME)
A1
B2
MATCH=I
35
ISIG='C'
B3
ISIG='D'
C3
VO=0.0
FLC=0.0
VC=0.0
B4
B3
TYPE MESSAGE
(INPUT NEW WGT OR VOLUME & CENTERS)
ACCEPT VO,FLC,VC
B4
ITYP=0
VO=VOX 2240.
ARRAY(1,MATCH)=VO
ARRAY(2,MATCH)=FLC
ARRAY(3,MATCH)=VC
ISIG≠'D'
B6
NEND=N-MATCH
N=N-1
B5
B5
DO 25 I=0,NEND
ARRAY(1,MATCH+I)=ARRAY(1,MATCH+I+1)
25
B6
TYPE MESSAGE
(ANY MORE CORRECTIONS)
ACCEPT ISIG
ISIG='Y'
A5
C1
B


C1
TYPE MESSAGE
(DO YOU WISH TO STORE)
ACCEPT ISIG
ISIG='N'
ISIG≠'Y'
RETURN
ITYP = 1
CALL ADDON (ARRAY,ITYP,3,N)
C2
C2
ITYP = 0
CALL ADDON (ARRAY,ITYP,2,N)
RETURN
C3
TYPE MESSAGE
(DETECT ERROR INPUT)
A1

IX-11.2 Listing of Subroutine UPDATE

```
      SUBROUTINE UPDATE(ARRAY,ITYP,N)
      DIMENSION ARRAY(3,100)
  1   TYPE 50
 50   FORMAT(' IS WEIGHT TO  BE UPDATED TO BE CHANGED, DELETED,'
     *  ' OR ADDED?'/' TYPE C,D, OR A -- ',$)
 42   ACCEPT 51,ISIG
 51   FORMAT(A1)
      IF(ISIG.EQ.'A')GO TO 30
      IF(ITYP.EQ.1)GO TO 10
C*******ITYP=1 UPDATE VOLUMES, ITYPE=0 UPDATE WEIGHTS.
      TYPE 54
 52   FORMAT(' INPUT PRESENT VOLUME, LCG, AND VCG -- ',$)
      ACCEPT 53, VO,FLC,VC
 53   FORMAT(3F)
      GO TO 20
 10   TYPE 52
 54   FORMAT(' INPUT PRESENT WEIGHT, LCG, AND VCG -- ',$)
      ACCEPT 53,VO,FLC,VC
      GO TO 20
C******* ADD NEW WEIGHT OR VOLUME TO DISTRIBUTION.
 30   IF(ITYP.EQ.1)GO TO 11
      TYPE 55
 55   FORMAT(' INPUT WEIGHT, LCG, VCG TO BE ADDED -- ',$)
      ACCEPT 53, VO,FLC,VC
      VO=VO*2240.
      GO TO 15
 11   TYPE 56
 56   FORMAT (' INPUT VOLUME , LCG, AND VCG TO BE ADDED -- ',$)
      ACCEPT 53,VO,FLC,VC
 15   N=N+1
      ARRAY(1,N)=VO
      ARRAY(2,N)=FLC
      ARRAY(3,N)=VC
      GO TO 100
C ******BEGIN SEARCH FOR CHANGE OR DELETE OPTION.
 20    DO 35 I=1,N
      IF(IFIX(ARRAY(1,I)*100.).NE.IFIX(VO*100.))GO TO35
      IF(IFIX(ARRAY(2,I)*100.).NE.IFIX(FLC*100.))GO TO 35
      IF(IFIX(ARRAY(3,I)*100.).NE.IFIX(VC*100.))GO TO 35
      MATCH=I
      GO TO 36
 35   CONTINUE
      TYPE 57
 57   FORMAT(' WEIGHT OR VOLUME AND POSITION CANNOT BE FOUND')
      GO TO 100
 36   IF(ISIG.EQ.'C')GO TO 40
      IF(ISIG.EQ.'D')GO TO 38
      TYPE 58
 58   FORMAT(' DETECT ERROR IN INPUT, TRY AGAIN')
      GO TO 1
```

```
38      VO=0.0
        FLC=0.0
        VC=0.0
        GO TO 37
40      TYPE 59
59      FORMAT(' INPUT NEW WGT OR VOLUME, AND LCG, VCG -- ',$)
        ACCEPT 53,VO,FLC,VC
        IF(ITYP.EQ.0) VO=VO*2240.
37      ARRAY(1,MATCH)=VO
        ARRAY(2,MATCH)=FLC
        ARRAY(3,MATCH)=VC
        IF(ISIG.NE.'D')GO TO 100
        NEND=N-MATCH
        N=N-1
        DO 25 I=0,NEND
        ARRAY(1,MATCH+I)=ARRAY(1,MATCH+I+1)
        ARRAY(2,MATCH+I)=ARRAY(2,MATCH+I+1)
        ARRAY(3,MATCH+I)=ARRAY(3,MATCH+I+1)
25     CONTINUE
100     TYPE 60
60      FORMAT(' DISTRIBUTION UPDATED, ANY MORE CORRECTIONS?'/
     *  ' TYPE Y OR N -- ',$)
        ACCEPT 51,ISIG
        IF(ISIG.EQ.'Y')GOTO 1
        TYPE 62
 62     FORMAT(' DO YOU WISH TO STORE THIS DISTRIBUTION FOR FUTURE USE'
     *  /' TYPE Y OR N -- '$)
        ACCEPT 51, ISIG
        IF(ISIG.EQ.'N')GO TO 43
        IF(ISIG.NE.'Y')GO TO 43
        IF(ITYP.EQ.1)CALL ADDON(ARRAY,ITYP,23,N)
        IF(ITYP.EQ.0)CALL ADDON(ARRAY,ITYP,21,N)
43      RETURN
        END
```

IX-12. Subroutine ASTMES

System messages in response to the monitor command, HELP), are read from a data file and printed out at the user's terminal through this subroutine. The subroutine opens the previously created file, reads 72 characters, checks the last character, the next to last character, the second from last character, etc., until it finds a non-blank character, then goes to a printing loop and prints one line up to and including the first non-blank character mentioned above.

The next 72 characters are then read and a similar procedure is followed until an end of file mark is encountered at which time control returns to the monitor.

There are no error messages.

Input Variables

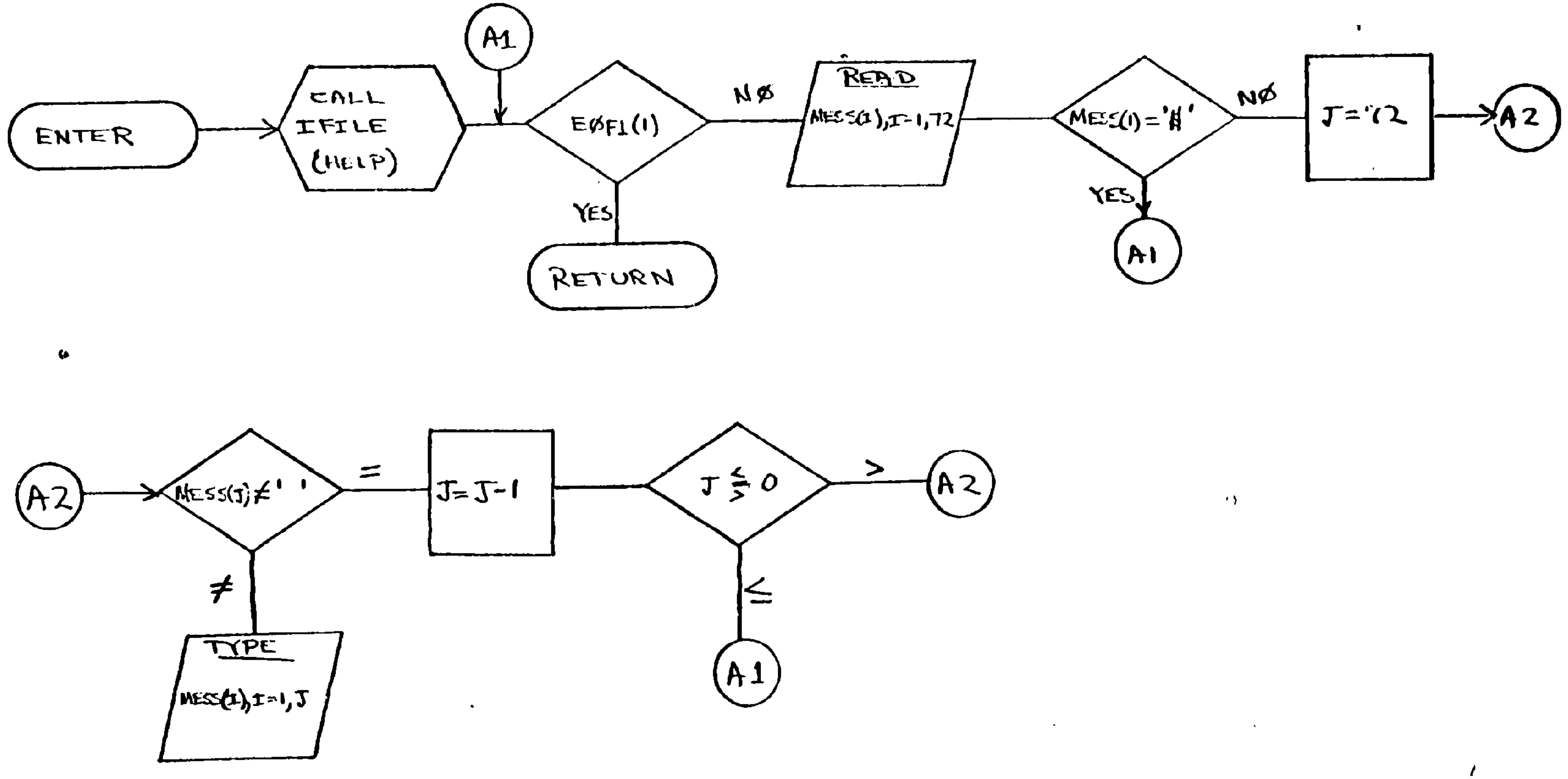
--None--

Output Variables

MESS	Message array containing one line of text

IX-12.1 Flow Chart for Subroutine ASTMES

SUBROUTINE ASTMES



IX-12.2 Listing of Subroutine ASTMES

```
      SUBROUTINE ASTMES
      DIMENSION MESS(72)
      CALL IFILE(1,'HELP')
1     IF (EOF1(1)) GOTO 99
      READ(1,10),(MESS(I),I=1,72)
10    FORMAT(72A1)
      IF(MESS(1).EQ.'$') GOTO 50
      J=72
2     IF(MESS(J).NE.' ') GOTO 20
      J=J-1
      IF(J) 1,1,2
20    TYPE 30,(MESS(I),I=1,J)
30    FORMAT(1H ,72A1)
      GOTO 1
50    TYPE 31
31    FORMAT(/)
      GO TO 1
99    RETURN
      END
```

IX-13. Subroutine RDVW

RDVW retrieves weight or volume distributions from their respective files if the commands INPUT WEIGHTS or INPUT VOLUMES is used and the option to use old weights or volumes is chosen. The user must input the identifier of the distribution he wishes to input from the file. When the distribution has been retrieved the message: "DISTRIBUTION IS NOW AVAILABLE FOR COMPUTATION AND HAS BEEN SAVED FOR FUTURE USE" is output, and control returns to the monitor.

Error Messages

1) "IDENTIFIER XXXXX NOT ON FILE, RETURN TO MONITOR" - The identifier XXXXX has been input incorrectly or that distribution does not exsist on file.

2) "ERROR IN DATA FILE, RETURN TO MONITOR" - The number of weights or volumes in the distribution is incorrect. The distribution should be reentered from the keyboard and the file updated.

Input Variables

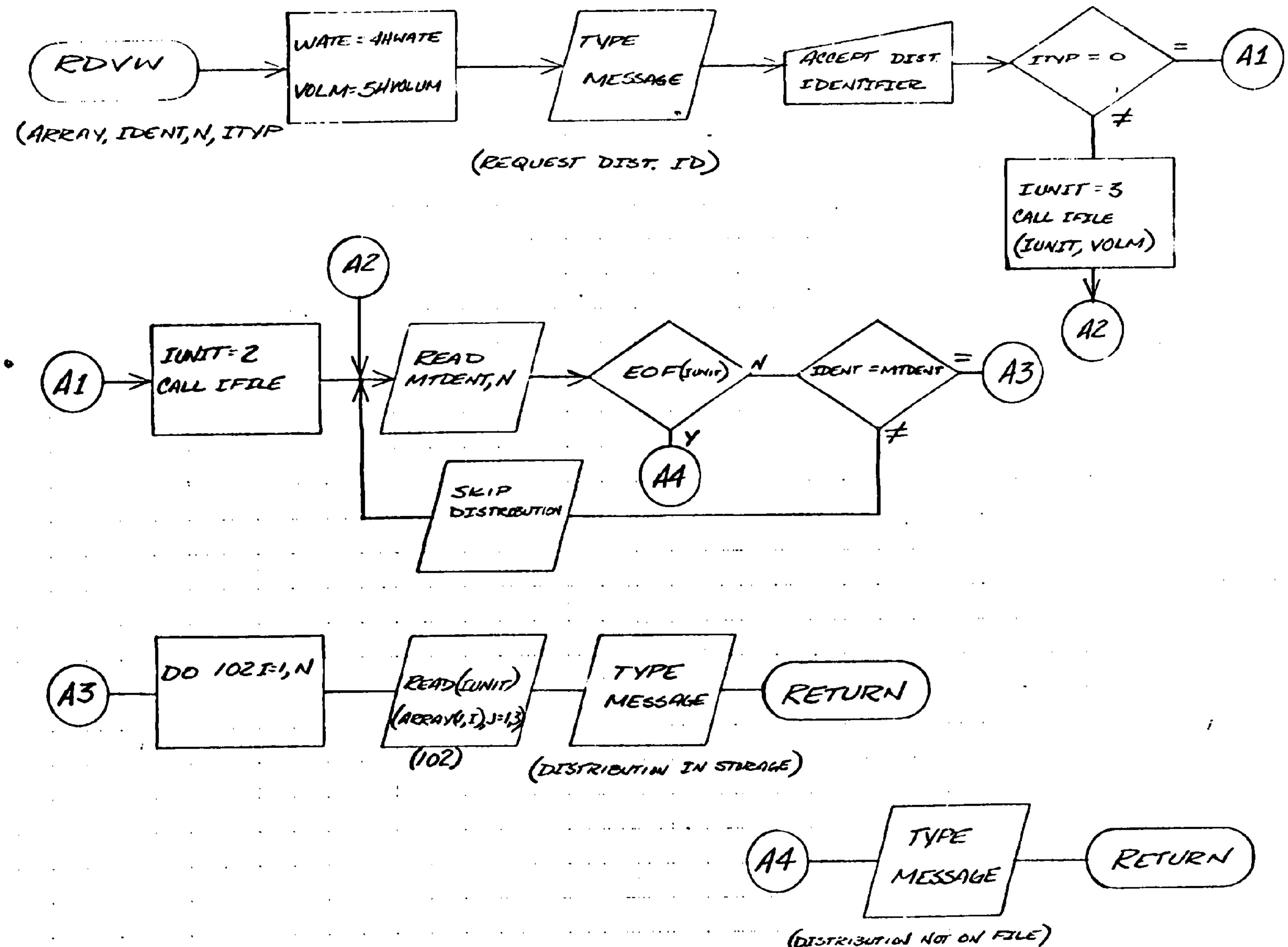
ARRAY	Array which retrieved distribution is to be stored in.
IDENT	Identifier of distribution to be retrieved
ITYP	Specifics whether weights or volumes are to be input

Output Variables

ARRAY	Array containing distribution
N	Number of triplets in distribution

IX-13.1 Flow Chart for Subroutine RDVW

SUBROUTINE RDVW



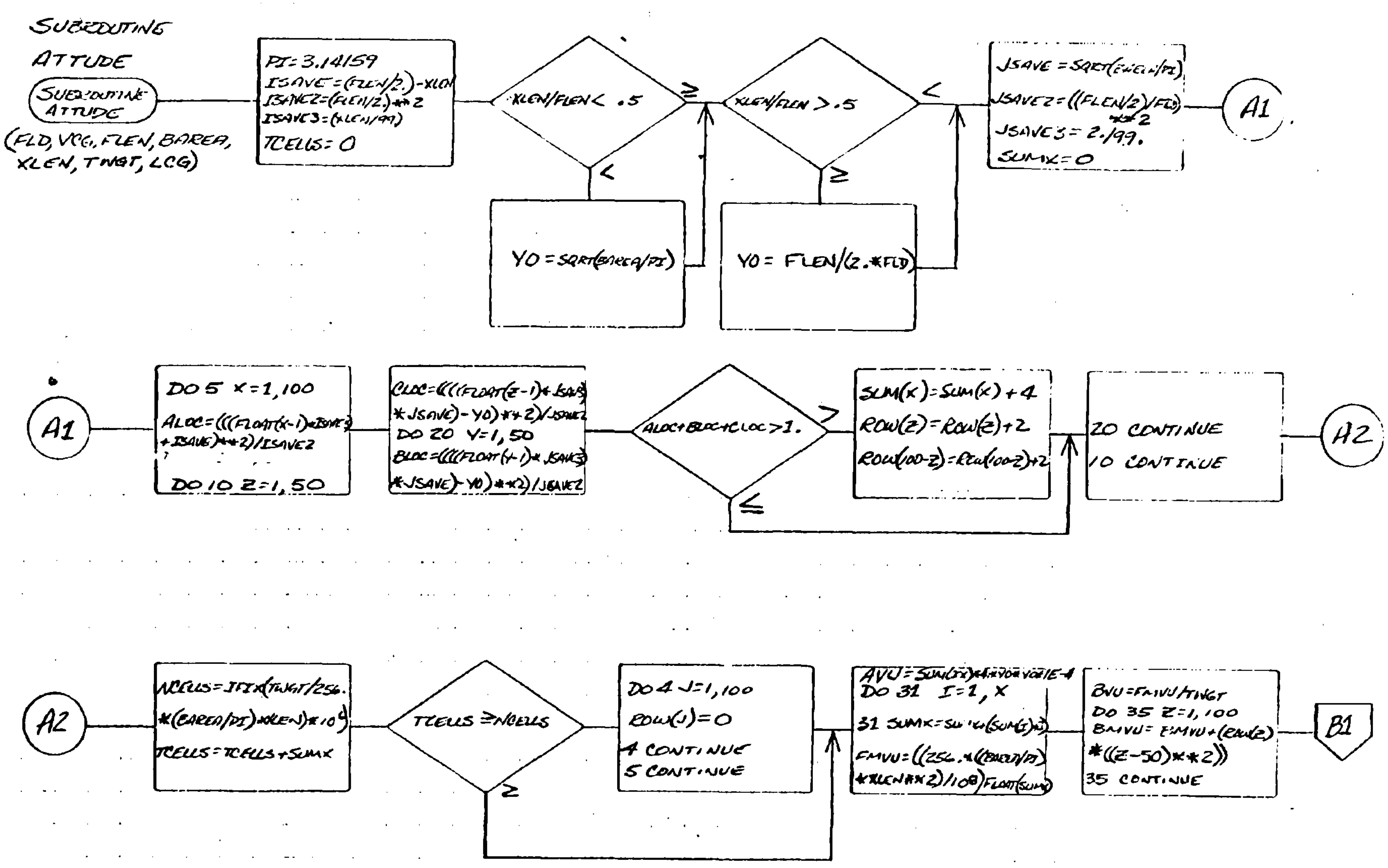
IX-13.2 Listing of Subroutine RDVW

```
      SUBROUTINE RDVW(ARRAY,IDENT,N,ITYP)
      DIMENSION ARRAY(3,100)
        WATE=4HWATE
       VOLM=5HVOLUM
51    FORMAT(A1)
       TYPE 53
53     FORMAT(' PLEASE INPUT DISTRIBUTION IDENTIFIER'/
     * ' 5 CHARACTERS ALLOWED -- '$)
      ACCEPT 52,IDENT
52    FORMAT(A5)
      IF(ITYP.EQ.0)GO TO 10
      IUNIT=23
      CALL IFILE(IUNIT,VOLM)
       GO TO 12
10    IUNIT=21
14    CALL IFILE(IUNIT,WATE)
12    READ(IUNIT,54) MTDENT,N
      IF(EOFC(IUNIT))GO TO 100
       IF(MTDENT.EQ.IDENT)GO TO 101
      DO 15 I=1,N
15    READ(IUNIT,51)DUMMY
      GO TO 12
101   DO 102 I=1,N
      READ(IUNIT,55)(ARRAY(J,I),J=1,3)
102   IF(EOFC(IUNIT))GO TO 99
      TYPE 56
55    FORMAT(3F)
54    FORMAT(A5,I3)
56    FORMAT(' DISTRIBUTION IS NOW AVAILABLE FOR COMPUTATION AND'/
     * ' HAS BEEN SAVED FOR FUTURE USE'/)
      RETURN
100   TYPE 57 ,IDENT
57    FORMAT(' IDENTIFIER 'A5,' NOT ON FILE, RETURN TO MONITOR')
       REWIND IUNIT
      RETURN
99    TYPE 58
58    FORMAT(' ERROR IN WEIGHT DATA FILE, RETURN TO MON'
     * 'ITOR')
      RETURN
      END
```

IX-14. Surface Motion Predection (WVE)

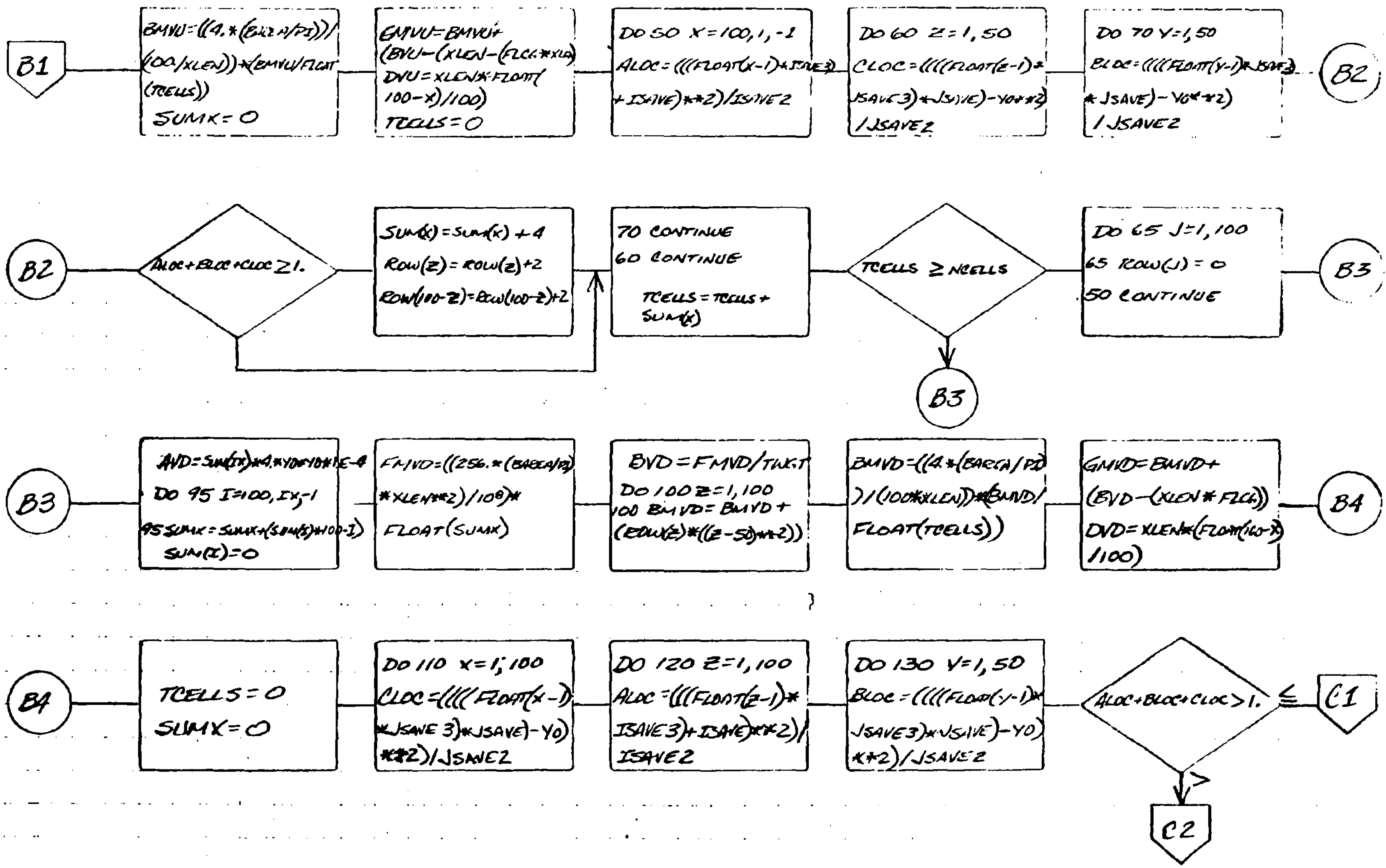
This subroutine calculates the heave and pitch response of the capsule as a function of the sea state. Pierson-Muskowitz fully developed, long-crested sea spectra are assumed. The capsule is assumed to float in one of three attitudes; vertical bow up, vertical bow down, or horizontal. The attitude which yields the largest metacentric height is selected as the most stable. Once the most stable attitude is selected the response is calculated. The subroutine prints out the attitude of the capsule on the surface. The subroutine also plots graphs of the pitch angle and heave amplitude as a function of wave length.

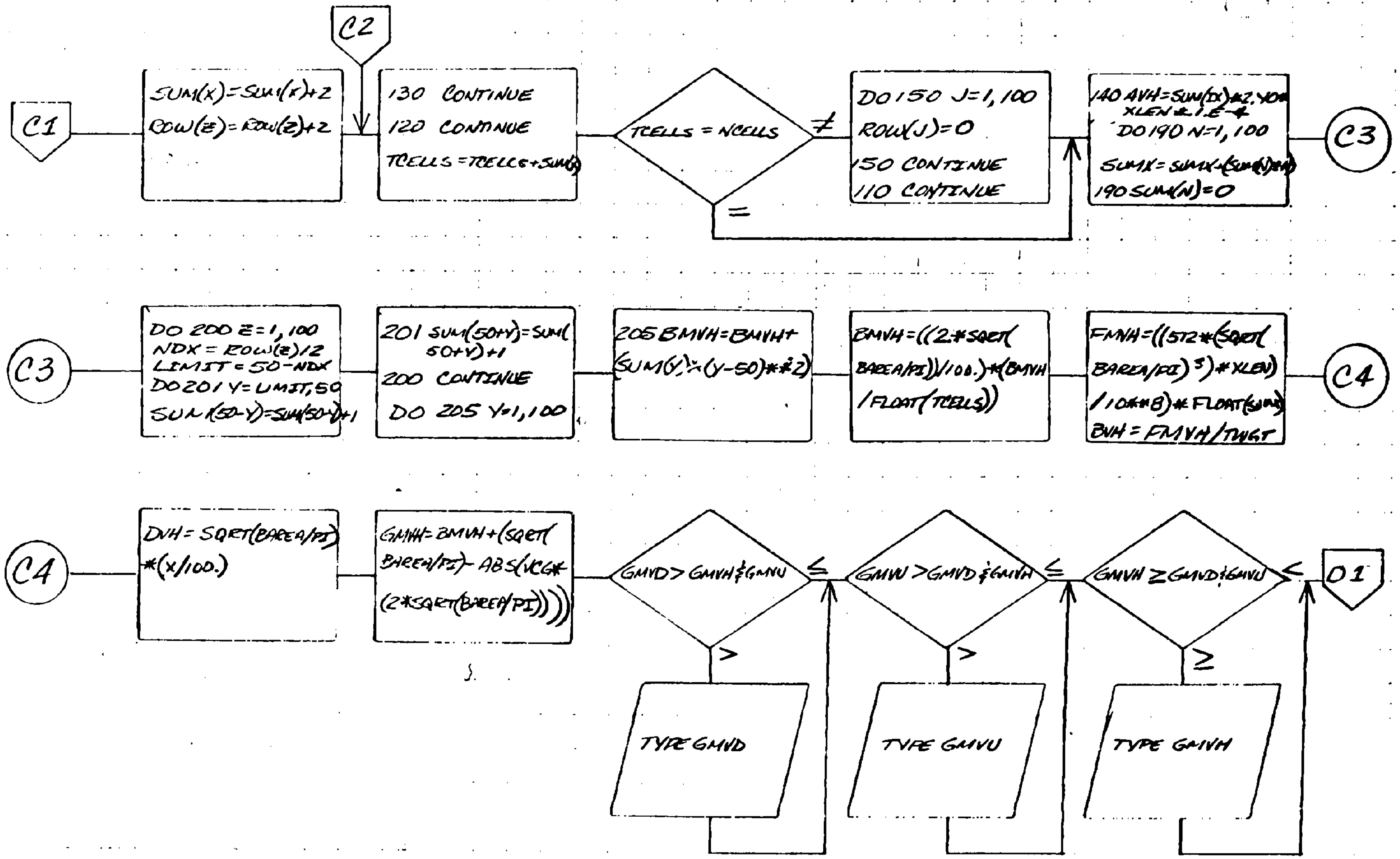
IX-14.1 Flow Chart for Subroutine WVE

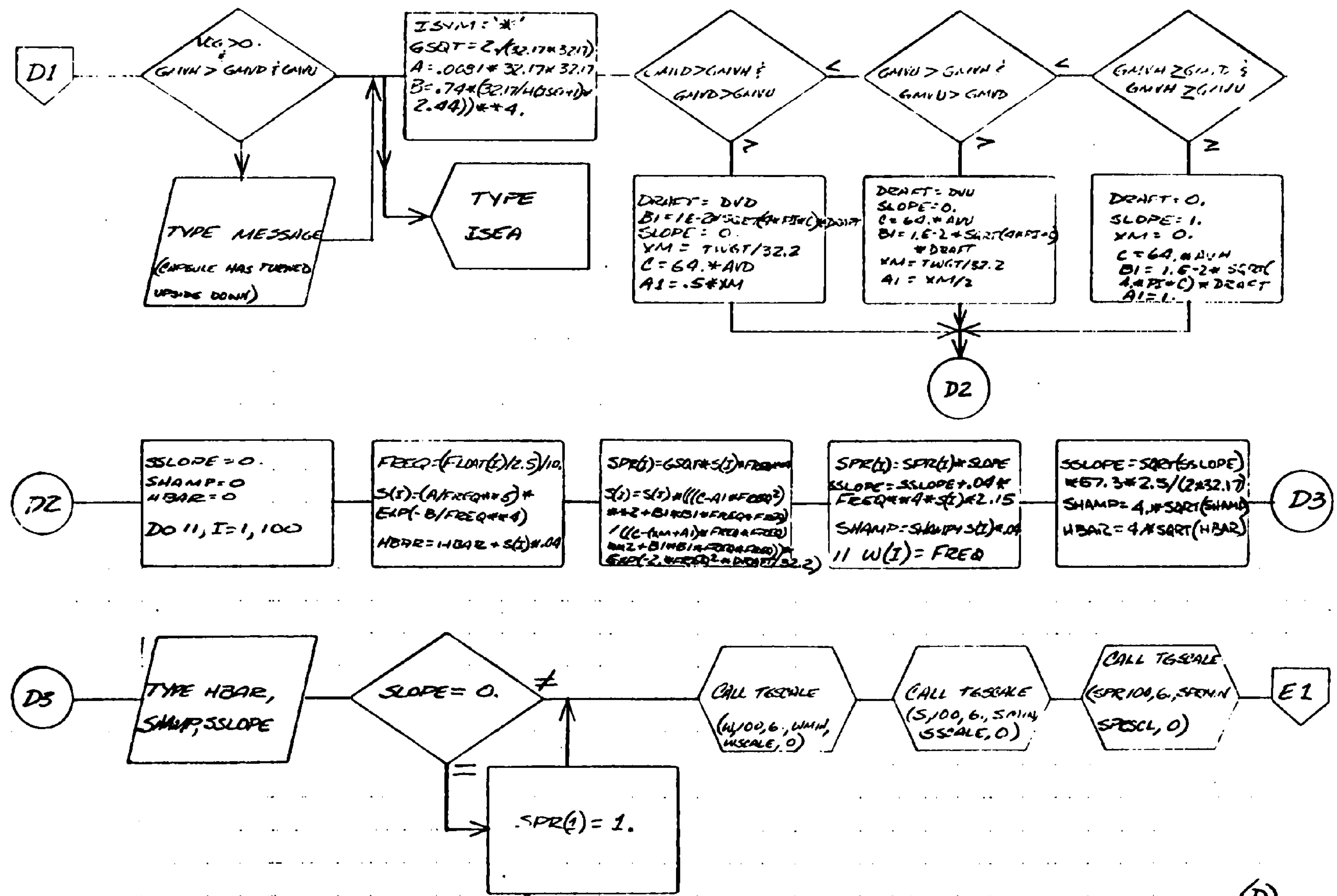


SUBROUTINE
ATTUDE
SUBROUTINE ATTUDE
(FLD, VCG, FLEN, BAREA, XLEN, TWGT, LCG)
PI=3.14159
ISAVE=(FLEN/2.)-XLEN
ISAVE2=(FLEN/2.)**2
ISAVE3=(XLEN/99)
TCELLS=0
XLEN/FLEN < .5
≥
<
YO=SQRT(BAREA/PI)
XLEN/FLEN > .5
<
≥
YO= FLEN/(2.*FLD)
JSAVE=SQRT(BAREA/PI)
JSAVE2=((FLEN/2)/FLD)**2
JSAVE3= 2./99.
SUMX=0
A1
A1
DO 5 X=1,100
ALOC=(((FLOAT(X-1)*ISAVE3)+ISAVE)**2)/ISAVE2
DO 10 Z=1,50
CLOC=(((FLOAT(Z-1)*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
DO 20 Y=1,50
BLOC=(((FLOAT(Y-1)*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
ALOC+BLOC+CLOC>1.
>
≤
SUM(X)=SUM(X)+4
ROW(Z)=ROW(Z)+2
ROW(100-Z)=ROW(100-Z)+2
20 CONTINUE
10 CONTINUE
A2
A2
NCELLS=IFIX(TWGT/256.*(BAREA/PI)*XLEN)*10^9
TCELLS=TCELLS+SUMX
TCELLS ≥ NCELLS
≥
DO 4 J=1,100
ROW(J)=0
4 CONTINUE
5 CONTINUE
AVU=SUM(X)
DO 31 I=1, X
31 SUMX=SUMX+(SUM(I)
FMVU=((256.*((BAREA/PI)*XLEN**2)/10^8)FLOAT(SUMX)
BVU=FMVU/TWGT
DO 35 Z=1,100
BMVU=FMVU+(ROW(Z)*((Z-50)**2))
35 CONTINUE
B1

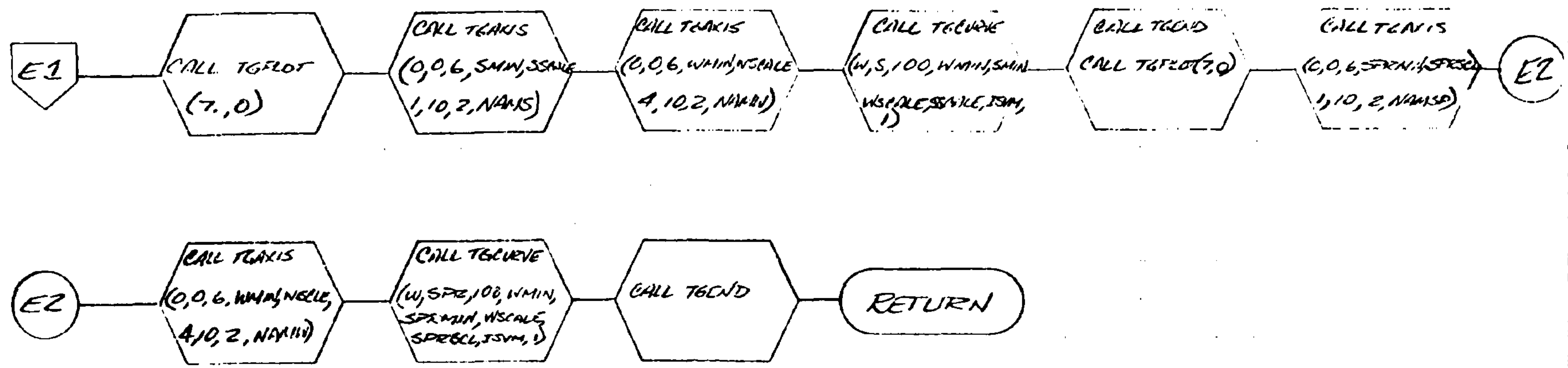
(A)


B1
SUMX=0
GMVU=BMVU+
DVU=XLEN*FLOAT(
100-X)/100)
TCELLS=0
DO 50 X=100,1,-1
DO 60 Z=1,50
/JSAVE2
DO 70 Y=1,50
*JSAVE)-YO**2)
/JSAVE2
B2
SUM(X)=SUM(X)+4
ROW(Z)=ROW(Z)+2
70 CONTINUE
60 CONTINUE
TCELLS=TCELLS+
SUM(X)
TCELLS ≥ NCELLS
DO 65 J=1,100
65 ROW(J)=0
50 CONTINUE
B3
FLOAT(SUMX)
BVD=FMVD/TWKT
DO 100 Z=1,100
100 BMVD=BMVD+
FLOAT(TCELLS))
GMVD=BMVD+
/100)
B4
TCELLS=0
SUMX=0
DO 110 X=1,100
DO 120 Z=1,100
ISAVE2
DO 130 Y=1,50
C1
C2
(B)


C1
SUM(X)=SUM(X)+2
ROW(Z)=ROW(Z)+2
C2
130 CONTINUE
120 CONTINUE
TCELLS=TCELLS+SUM(X)
TCELLS = NCELLS
≠
=
DO 150 J=1,100
ROW(J)=0
150 CONTINUE
110 CONTINUE
140 AVH=SUM(IX)*2.*XLEN*1.E-4
DO 190 N=1,100
SUMX=SUMX+(SUM(N)*N)
190 SUM(N)=0
C3
C3
DO 200 Z=1,100
NDX=ROW(Z)/2
LIMIT=50-NDX
DO 201 Y=LIMIT,50
SUM(50-Y)=SUM(50-Y)+1
201 SUM(50+Y)=SUM(50+Y)+1
200 CONTINUE
DO 205 Y=1,100
205 BMVH=BMVH+SUM(Y)*(Y-50)**2
BMVH=((2.*SQRT(BAREA/PI))/100.)*(BMVH/FLOAT(TCELLS))
FMVH=((512*(SQRT(BAREA/PI))**3)*XLEN)/10**8)*FLOAT(SUMX)
BVH=FMVH/TWGT
C4
C4
DVH=SQRT(BAREA/PI)*(X/100.)
GMVH=BMVH+(SQRT(BAREA/PI)-ABS(VCG*(2*SQRT(BAREA/PI))))
GMVD > GMVH & GMVU
>
≤
TYPE GMVD
GMVU > GMVD & GMVH
>
≤
TYPE GMVU
GMVH ≥ GMVD & GMVU
≥
<
TYPE GMVH
D1


D1
TYPE MESSAGE (CAPSULE HAS TURNED UPSIDE DOWN)
TYPE ISEA
A = .0081 * 32.17 * 32.17
DRAFT = DVD
SLOPE = 0.
YM = TWGT/32.2
C = 64. * AVD
A1 = .5 * YM
DRAFT = DVU
SLOPE = 0.
C = 64. * AVU
YM = TWGT/32.2
A1 = YM/2
DRAFT = 0.
SLOPE = 1.
YM = 0.
C = 64. * AVH
D2
D2
SSLOPE = 0.
SHAMP = 0
HBAR = 0
DO 11, I = 1, 100
HBAR = HBAR + S(I) * .04
SHAMP = SHAMP + S(I) * .04
11 W(I) = FREQ
SHAMP = 4. * SQRT(SHAMP)
HBAR = 4. * SQRT(HBAR)
D3
D3
TYPE HBAR, SHAMP, SSLOPE
SLOPE = 0.
SPR(1) = 1.
CALL TGSCALE (W, 100, 6., WMIN, WSCALE, 0)
CALL TGSCALE (S, 100, 6., SMIN, SSCALE, 0)
CALL TGSCALE (SPR, 100, 6., SPRMIN, SPRSCL, 0)
E1

(D)


E1
CALL TGPLOT (7.,0)
CALL TGAXIS (0,0,6,SMIN,SSCALE, 1,10,2,NAXIS)
CALL TGAXIS (0,0,6,WMIN,WSCALE, 4,10,2,NAXIW)
CALL TGCURVE (W,S,100,WMIN,SMIN, WSCALE,SSCALE,ISYM, 1)
CALL TGEND
CALL TGPLOT(7,0)
CALL TGAXIS
E2
E2
CALL TGAXIS
CALL TGCURVE
CALL TGEND
RETURN

IX-14.2 Listing of Subroutine WVE

```
      SUBROUTINE WVE(FLD,VCG,FLEN,BAREA,XLEN,TWGT,FLCG,ISEA)
      DIMENSION H(10),NAMW(12),NAMS(12),NAMP(12),NAMSP(12),
     *NAMPP(12)
         DIMENSION S(100),SPR(100),W(100)
      DATA(H(I),I=1,10)/.12,.85,2.25,4.25,6.75,10.,18.,
     *32.5,52.5,85./
      DATA(NAMW(K),K=1,12)/'FREQUENCY, RAD/SEC  ',8*'    '/
      DATA(NAMS(K),K=1,12)/'HEAVE AMPL SQRD/FREQ',8*'    '/
      DATA(NAMP(K),K=1,12)/'(HEAVE AMPLITUDE)**2/FREQ VS. FREQ ',
     *5*'    '/
      DATA(NAMSP(K),K=1,12)/'PITCH ANGLE SQRD/FREQ    ',7*'    '/
      DATA(NAMPP(K),K=1,12)/'PITCH ANGLE SQRD/FREQ    ',7*'    '/
      DIMENSION SUM(100),ROW(100)
      DIMENSION MESS(3,3)
      DATA((MESS(I,J),J=1,3),I=1,3)/' BOW ','DOWN.','     ',' BOW ',
     * 'UP.  ','     ',' HORI','ZONTA','LLY. '/
      REAL ISAVE,ISAVE2,ISAVE3,JSAVE,JSAVE2,JSAVE3
       INTEGER X,Y,Z,ROW,SUM,TCELLS,SUMX
      DO 301 I=1,100
  301 SUM(I)=0
      CALL ERRSET(0)
      PI=3.14159
C     COMPUTE CONSTANT PORTIONS OF EQUATION.
      ISAVE=(FLEN/2.)-XLEN
      ISAVE2=(FLEN/2.)**2
      ISAVE3=XLEN/99.
      IF(XLEN/FLEN.LT..5)YO=SQRT(BAREA/PI)
      IF(XLEN/FLEN.GE..5)YO=FLEN/(2.*FLD)
      JSAVE=YO
      JSAVE2=((FLEN/2)/FLD)**2
      JSAVE3=2./99.
      SUMX=0
      TCELLS=0
      NCELLS=IFIX(TWGT/(256.*(YO*YO)*XLEN)*10.**6.)
C     FIND REGION OCCUPIED BY CAPSULE(VERTICAL ATTITUDE)
      DO 5 X=1,100
      IX=X
      SX=X-1
       ALOC=(((SX*ISAVE3)+ISAVE)**2)/ISAVE2
      DO 10 Z=1,50
      SZ=Z-1
       CLOC=((((SZ*JSAVE3)*JSAVE)-YO)**2)/ISAVE2
      SAVLOC=ALOC+CLOC
      DO 20 Y=1,50
      SY=Y-1
       BLOC=((((SY*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
      IF(ALOC+BLOC+CLOC.GT.1.)GO TO 20
      SUM(X)=SUM(X)+4
      ROW(Z)=ROW(Z)+2
```

```
      ROW(100-Z)=ROW(100-Z)+2
 20   CONTINUE
 10   CONTINUE
      TCELLS=TCELLS+SUM(X)
      IF(TCELLS.GE.NCELLS)GO TO 30
      DO 4 J=1,100
 4    ROW(J)=0
 5    CONTINUE
 30   AVU=SUM(IX)*4.*    YO*1.E-4
      DO 31 I=1,IX
      SUMX=SUMX+(SUM(I)*(I-1))
 31   SUM(I)=0
      FMVU=((256.*((YO*YO))*XLEN**2.)/10.**8.)*FLOAT(SUMX)
      BVU=FMVU/TWGT
      DO 35 Z=1,100
      BMVU=BMVU+(ROW(Z)*((Z-50)**2))
 35    CONTINUE
     BMVU=((4.*(YO*YO))/(100.*XLEN))*(BMVU/FLOAT(TCELLS))
      GMVU=BMVU+(BVU-(XLEN-(FLCG*XLEN)))
      DVU=XLEN*(FLOAT(X)/100.)
      TCELLS=0
      SUMX=0
C     COMPUTE BOW DOWN ATTITUDE.
      DO 50 X=100,1,-1
      IX=X
      SX=X-1
      ALOC=(((SX*ISAVE3)+ISAVE)**2)/ISAVE2
      DO 60 Z=1,50
      SZ=Z-1
      CLOC=((((SZ*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
      SAVLOC=ALOC+CLOC
      DO 70 Y=1,50
      SY=Y-1
      BLOC=((((SY*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
      IF(SAVLOC+BLOC.GT.1.)GO TO 70
      SUM(X)=SUM(X)+4
      ROW(Z)=ROW(Z)+2
      ROW(100-Z)=ROW(100-Z)+2
 70   CONTINUE
 60   CONTINUE
      TCELLS=TCELLS+SUM(X)
      IF(TCELLS.GE.NCELLS)GO TO 90
      DO 65  J=1,100
 65     ROW(J)=0
 50   CONTINUE
 90   AVD=SUM(IX)*4.*YO*YO*1.E-4
     DO 95 I=100,IX,-1
      SUMX=SUMX+(SUM(I)*(100-I))
 95   SUM(I)=0
      FMVD=((256.*(YO*YO)*XLEN**2.)/10.**8.)*FLOAT(SUMX)
      BVD=FMVD/TWGT
```

```
      DO 100 Z=1,100
100   BMVD=BMVD+(ROW(Z)*((Z-50)**2))
      BMVD=((4.*(YO*YO))/(100.*XLEN))*(BMVD/FLOAT(TCELLS))
      GMVD=BMVD+(BVD-(XLEN*FLCG))
      DVD=XLEN*(FLOAT(100-X)/100.)
            TCELLS=0
      SUMX=0
C     FIND AREAS IN HORIZONTAL ATTITUDE.
      DO 110 X=1,100
      IX=X
      SX=X-1
      CLOC=((((SX*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
      DO 120 Z=1,100
      SZ=Z-1
      ALOC=(((SZ*ISAVE3)+ISAVE)**2)/ISAVE2
      SAVLOC=ALOC+CLOC
      DO 130 Y=1,50
      SY=Y-1
      BLOC=((((SY*JSAVE3)*JSAVE)-YO)**2)/JSAVE2
      IF(SAVLOC+BLOC.GT.1.)GO TO 130
      SUM(X)=SUM(X)+2
      ROW(Y)=ROW(Y)+2
130   CONTINUE
120   CONTINUE
      TCELLS=TCELLS+SUM(X)
      IF(TCELLS.GE.NCELLS)GOTO 140
      DO 150 J=1,50
      ROW(J)=0
150     CONTINUE
110     CONTINUE
140   AVH=SUM(IX)*2.*YO*XLEN*1.E-4
      DO 190 N=1,IX
      SUMX=SUMX+(SUM(N)*(N-1))
190      SUM(N)=0.
       DO 205 Y=1,50
205    BMVH=BMVH+(ROW(Y)*(Y-50)**2)
       BMVH=((2.*YO)/100.)*(BMVH/FLOAT(TCELLS))
       FMVH=((512.*(YO**3.)*XLEN)/10**8)*FLOAT(SUMX)
       BVH=FMVH/TWGT
      DVH=(2.*YO)*(FLOAT(X)/100.)
700   CONTINUE
      IF(GMVD.GT.GMVH.AND.GMVD.GT.GMVU)TYPE 300,(MESS(1,I),I=1,3)
      IF(GMVU.GT.GMVD.AND.GMVU.GT.GMVH)TYPE 300,(MESS(2,I),I=1,3)
      IF(GMVH.GE.GMVD.AND.GMVH.GE.GMVU)TYPE 300,(MESS(3,I),I=1,3)
      IF(VCG.GT.0..AND.GMVH.GT.GMVD.AND.GMVH.GT.GMVU)TYPE 400
300   FORMAT(//' THE CAPSULE IS FLOATING',3A5,/)
400   FORMAT(' AND HAS TURNED UPSIDE DOWN',//)
      ISYM='*'
```

```
      TYPE 302,ISEA
 302  FORMAT(' SURFACE SEA STATE IS ',I2,/)
      GSQT=2./(32.17*32.17)
      A=.0081*32.17*32.17
      B=.74*(32.17/(H(ISEA+1)*2.44))**4.
      IF(GMVD.GT.GMVH.AND.GMVD.GT.GMVU)GO TO 500
      IF(GMVU.GT.GMVH.AND.GMVU.GT.GMVD)GO TO 510
      IF(GMVH.GE.GMVD.AND.GMVH.GE.GMVU)GO TO 520
 500  DRAFT=DVD
      SLOPE=0.
      C=64.*AVD
      B1=1.E-2*SQRT(4.*PI*C)*DRAFT
      XM=TWGT/32.2
      A1=.5*XM
      GO TO 530
 510  DRAFT=DVU
      SLOPE=0.
      C=64.*AVU
      B1=1.E-2*SQRT(4.*PI*C)*DRAFT
      XM=TWGT/32.2
      A1=XM/2.
      GO TO 530
 520  DRAFT=0.
      SLOPE=1.
      XM=0.
      C=64.*AVH
      B1=1.E-2*SQRT(4.*PI*C)*DRAFT
      A1=1.
 530  SSLOPE=0.
      SHAMP=0.
      HBAR=0.
      DO 11 I=1,100
      FREQ=(FLOAT(I)/2.5)/10.
      S(I)=(A/FREQ**5)*EXP(-B/FREQ**4)
      HBAR=HBAR+S(I)*.04
      SPR(I)=GSQT*S(I)*FREQ**4
      S(I)=S(I)*(((C-A1*FREQ*FREQ)**2+B1*B1*FREQ*FREQ)/
     * ((C-(XM+A1)*FREQ*FREQ)**2+B1*B1*FREQ*FREQ))*
     * EXP(-2.*FREQ*FREQ*DRAFT/32.2)
      SPR(I)=SPR(I)*SLOPE
      SSLOPE=SSLOPE+.04*FREQ**4.*S(I)*2.15
      SHAMP=SHAMP+S(I)*.04
 11     W(I)=FREQ
      SSLOPE=SQRT(SSLOPE)*57.3*2.5/(2.*32.17)
      SHAMP=4.*SQRT(SHAMP)
      HBAR=4.*SQRT(HBAR)
      ..PE 303,HBAR,SHAMP,SSLOPE
 303  FORMAT(' SIGNIFICANT WAVE HEIGHT = ',F10.3,' FT',/
     *        ' SIGNIFICANT HEAVE AMPL. = ',F10.3,' FT',/
     *        ' AVERAGE PITCH AMPL.     = ',F10.3,' DEGREES',/)
```

```
      IF(SLOPE.EQ.0.)SPR(1)=1.
     CALL TGSCALE(W,100,6.,WMIN,WSCALE,0)
      CALL TGSCALE(S,100,6.,SMIN,SSCALE,0)
      CALL TGSCALE(SPR,100,6.,SPRMIN,SPRSCL,0)
      CALL TGPLOT(7.,0)
      CALL TGAXIS(0.,0.,6.,SMIN,SSCALE,1,10,2,NAMS)
      CALL TGAXIS(0.,0.,6.,WMIN,WSCALE,4,10,2,NAMW)
      CALL TGCURVE(W,S,100,WMIN,SMIN,WSCALE,SSCALE,ISYM,1)
      CALL TGEND
      CALL TGPLOT(7.,0)
      CALL TGAXIS(0.,0.,6.,SPRMIN,SPRSCL,1,10,2,NAMSP)
      CALL TGAXIS(0.,0.,6.,WMIN,WSCALE,4,10,2,NAMW)
      CALL TGCURVE(W,SPR,100,WMIN,SPRMIN,WSCALE,SPRSCL,
     *ISYM,1)
      CALL TGEND
      RETURN
      END
```

IX-15 Data Files

The messages accessed in response to the monitor command HELP reside in a data file called HELP.DAT

Permanent storage of weight distributions and volume distributions is provided by the data files WATE.DAT and VØLUM.DAT, respectively. Since these files must contain some "data" in order to exist, an identification name, A0001, and a single null triplet have been stored in each file.

These data files are called automatically during loading (provided they exist) and therefore are not included in the LOAD or EXECUTE command string.

A listing of the contants of all three data files is shown on the following pages.

```
** ABBREVIATED USER INSTRUCTIONS:
$
THE PROGRAM YOU ARE RUNNING WILL HELP YOU DESIGN AN ELLIPSOIDAL SUB-
MARINE WITH A SEPARABLE CAPSULE.  EACH COMMAND BELOW EITHER PERFORMS A
COMPLETE FUNCTION, PROMPTS ADDITIONAL QUESTIONS FROM THE PROGRAM, OR PRO
-
VIDES YOU WITH INFORMATION.  THE PROGRAM SIGNALS THAT IT IS READY FOR A
COMMAND BY PRINTING  **  AT THE LEFT MARGIN OR FOR A RESPONSE BY
PRINTING  --  .
$
$
COMMAND                 DESCRIPTION
$
HELP                    PRINTS USER INSTRUCTIONS
$
INPUT GEOMETRY          LETS USER SPECIFY GENERAL GEOMETRIC CHARAC-
                        TERISTICS OF VEHICLE
$
INPUT WEIGHTS           LETS USER SPECIFY WEIGHT DISTRIBUTION OF VEHICLE
$
INPUT VOLUMES           LETS USER SPECIFY FLOODABLE VOLUME DISTRIBUTION
$
INPUT DEPTH             LETS USER SPECIFY INITIAL DEPTH
$
INPUT SPEED             LETS USER SPECIFY INITIAL VERTICAL AND HORIZONTAL
                        VELOCITIES
$
INPUT TRIM              LETS USER SPECIFY INITIAL TRIM ANGLE
$
INPUT SEAS              LETS USER SPECIFY SURFACE SEA STATE
$
UPDATE                  ALLOWS CHANGES TO BE MADE TO WEIGHT OR VOLUME
                        DISTRIBUTIONS
$
WEIGHTS                 CALCULATES WEIGHTS AND CENTERS
$
DAMAGE                  SIMULATES SELECTIVE FLOODING OF VOLUMES
$
TRIM                    CALCULATES BOUYANT FORCE AND TRIM OF INTACT
                        VEHICLE
$
SEPARATE                SIMULATES CAPSULE SEPARATION FROM HULL
$
ASCEND                  CALCULATES CAPSULE TRAJECTORY
$
```

```
HULL                    CALCULATES REMAINING HULL TRAJECTORY
$
GRAPH                   PLOTS BEHAVIOR OF SEPARATED CAPSULE OR REMAINING
                        VEHICLE (WHICHEVER HAD THE MORE RECENT TRAJECTORY
                        COMPUTED)
$
MOTION                  DETERMINES THE ATTITUDE OF CAPSULE ON
                        THE SURFACE AND PLOTS THE SEA SURFACE SPECTRUM
$
PRINT                   PRINTS TRAJECTORY DATA IN TABULAR FORM
$
CHECK                   PRINTS CURRENT STATUS OF SIMULATION
$
NEW CASE                INITIALIZES PROGRAM FOR A NEW SIMULATION
$
$
FOR ADDITIONAL DETAILS, CONSULT PROGRAM DOCUMENTATION
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) CADCOM, Incorporated 2024 West Street, Annapolis, MD 21401	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP NA

3. REPORT TITLE

SIMULATION STUDY OF A MODULAR ESCAPE CAPSULE RELEASED FROM A SUBMERGED SUBMARINE

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

FINAL 1 Feb. 1970—30 Nov. 1970

5. AUTHOR(S) (First name, middle initial, last name)

John C. Gebhardt
Frederick A. Klappenberger

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
30 Nov. 1970		

8a. CONTRACT OR GRANT NO. N00014-70-C-0247 b. PROJECT NO. NR277-061 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 70-2 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Office of NAVAL RESEARCH.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	ONR Arlington, VA

13. ABSTRACT

The hydrodynamic behavior of a submarine escape capsule during the separation and ascent phase was analyzed and simulated in a real-time digital system called MODSEC - MODular Submarine with Escape Capsule. The MODSEC system can simulate the behavior of a submarine with a forward escape capsule as the submarine sustains damage, the capsule separates, ascends to the surface, and floats on the surface awaiting rescue. Variables under user control are length/diameter ratio, length of submarine, length of capsule, casualty size and location, initial velocity, initial trim angle, sea state, weight distribution.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
SUBMARINE, ESCAPE, CAPSULE, MODULAR, HYDRODYNAMICS, SIMULATION, COMPUTER, FLOODABLE LENGTH, GEOMETRY, WEIGHTS, SURFACE, MONITOR, MOTIONS, STUDY, CASUALTY						